CICS-WINDOWS®

Technical Reference Guide

Release 4.2

Publication Number

GP50-0420-2

WORLD HEADQUARTERS
UNICOM Systems, Inc.
1032 Cove Way
Beverly Hills, CA 90210
(818) 838-0326

MISSION HILLS DIVISION
UNICOM Systems, Inc.
15535 San Fernando Mission Blvd.
Mission Hills, CA 91345
(818) 838-0606 Fax: (818) 838-07760

FAR EAST DIVISION
UNICOM Systems, Korea
Hawangshimnidong 301 Suite 102
Sungdong-ku, Seoul, Korea
(02) 296-5476 Fax: (02) 294-5446

Ninth Edition - June 2004

The information in this document applies to Release 4.2
of the CICS-WINDOWS® product for all operating environments.

Published by UNICOM Systems, Inc.
15535 San Fernando Mission Blvd., Suite 310
Mission Hills, CA 91345
(818) 838-0606, Fax: (818) 838-0776

© Copyright 1995-2004 UNICOM Systems, Inc.

CICS-WINDOWS® is an exclusively licensed trademark from Microsoft Corporation. *All rights reserved*

CICS-WINDOWS

Release 4.2.0

TABLE OF CONTENTS

PREFACE XI

SECTION 1. INTRODUCTION 15

CICS-WINDOWS -- A powerful productivity tool. 15

TERMINOLOGY 16

PRODUCT DEFINITION 18

The CICS-WINDOWS Demonstration Program 18
RUNNING THE DEMONSTRATION PROGRAM 18

SECTION 2. USING APPLICATIONS WITH CICS-WINDOWS ACTIVE 19

TUTORIAL 19
TOGGLE BACKWARD KEY OPERATION 22
DIRECT SESSION KEY OPERATION 22

SECTION 3. USING THE WINDOWING FEATURE 23

ENTERING WINDOW MODE 23
OPERATION WITH WINDOWING 23
SCROLLING COMMANDS 24
WINDOW SCROLLING KEY OPERATION 25
SWITCHING TO FULL-SCREEN MODE 26
TOGGLING WHILE IN WINDOW MODE 26
WINDOW ADJUSTMENT COMMANDS 28
WINDOW ADJUSTMENT RULES: 29
DESIGNATING A DOMINANT WINDOW 29
WINDOW SWITCH KEY OPERATION 31
CHANGING WINDOW CONFIGURATION 31
STANDARD, VARIABLE AND POPUP WINDOW MODES 31
STANDARD WINDOW CONFIGURATIONS 31
OPERATION WITH STANDARD WINDOWS 35
RESPONDING TO A WINDOW - STANDARD WINDOWS 36
CLEARING A WINDOW - STANDARD WINDOW 36
DIRECT SESSION KEY OPERATION IN STANDARD WINDOW MODE 36
VARIABLE WINDOW CONFIGURATIONS 37
RESPONDING TO A WINDOW - VARIABLE WINDOWS 38
OVERLAPPING WINDOWS 39
CLEARING A WINDOW - VARIABLE WINDOWS 39
DIRECT SESSION KEY OPERATION IN VARIABLE WINDOW MODE 39
METHODS OF OPERATION USING VARIABLE WINDOWS 39
OPERATION WITH POPUP WINDOWS 39
ACTION BAR FUNCTIONS OF POPUP WINDOWS 40
WINDOW EXAMPLES 41

SECTION 4. EXAMPLES OF USE 49

Example 1: Program development and testing. 49
Example 2: Having the Menu Available on the Same Screen as the Transaction 49

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Example 3: Using Data from Multiple Applications. 49
Example 4: Data Entry Exiting....49
Example 5: Saving your place in a multi-system environment. 49
Example 6: Toggling and Windowing Applications on Multiple CICS Systems using MRO or ISC 50
Summary 50

SECTION 5. OPERATION 51

INITIATING CICS-WINDOWS 51
THE CICS-WINDOWS USER CONFIGURATION DISPLAY 51
FIELDS OF THE USER CONFIGURATION DISPLAY....52
ENTERING COMMANDS AT THE USER CONFIGURATION DISPLAY 57
THE USER FUNCTION KEY ASSIGNMENT DISPLAY....58
FUNCTION KEY ASSIGNMENTS....58
TOGGLE AND WINDOW SWITCH COMMANDS....58
WINDOW MODE COMMANDS 59
CUT AND PASTE COMMANDS 59
CHANGING THE NUMBER OF WINDOWS 59
TERMINATING CICS-WINDOWS 60
THE CICS-WINDOWS SYSTEM DISPLAY....61
FIELDS OF THE SYSTEM DISPLAY 61
DISPLAYING THE PSEUDO TERMINAL IDS 62
DISPLAYING DATA COMPRESSION STATISTICS....62
VIEWING ANOTHER TERMINAL SESSION....63
USING THE CONTROL CHARACTER TO PERFORM COMMANDS....64
OPERATION OF CICS-WINDOWS HELP 65
THE CICS-WINDOWS CONTROL WINDOW....66
OPERATION OF ACTION BARS....67
INITIATING AN ACTION THROUGH A PULL-DOWN MENU 67
INITIATING AN ACTION THROUGH THE FAST PATH METHOD 67

USING APPLICATIONS WITH CICS-WINDOWS ACTIVE 68

SECTION 6. USING THE CUT AND PASTE FEATURE....69

COMMANDS USED IN CUT AND PASTE 69
THE COPY COMMAND 70
THE STACK COMMAND....70
THE PASTE COMMAND....70
THE UNPROTECT COMMAND 71

TYPES OF CUT AND PASTE OPERATIONS....72

SINGLE-FIELD CUT/PASTE....72
MULTIPLE-FIELD CUT/PASTE....72
FIELD PROPAGATION CUT/PASTE 73
LINE CUT/PASTE....73
FULL SCREEN CUT/PASTE....74
EXAMPLES OF CUT/PASTE OPERATIONS 76
SINGLE FIELD CUT/PASTE EXAMPLE....76
MULTIPLE FIELD CUT/PASTE EXAMPLE 79
FIELD PROPAGATION CUT/PASTE EXAMPLE 81
FIELD CUT/PASTE EXAMPLE USING THE STACK COMMAND....83
LINE CUT/PASTE EXAMPLE USING THE UNP COMMAND 85

SECTION 7. THE HELP-WINDOWS FEATURE....88

PREPARING TO DEFINE A HELP DISPLAY....88
DEFINING THE HELP KEY 89
DEFINING THE HELP-DEFINITION (DEFINE) KEY 89

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

AN ALTERNATE METHOD....90
ENTERING AUTO-DEFINE MODE....90
CREATING A FIELD LEVEL HELP DISPLAY, A QUICK OVERVIEW....91
STEP 1. INVOKING THE USER TRANSACTION....91
STEP 2. POSITIONING THE CURSOR....91
STEP 3. PRESS THE HELP-DEFINITION HOT KEY....92
STEP 4. CHANGING THE WINDOW....93
STEP 5. IDENTIFY THE HELP FIELD....94
STEP 6. IDENTIFY THE SCREEN....95
STEP 7. ENTER THE HELP TEXT....96
STEP 8. EXIT AUTO-DEFINE MODE....97
STEP 9. TEST THE HELP DISPLAY....98
CREATING A SCREEN LEVEL HELP DISPLAY....98
CREATING A TRANSACTION LEVEL HELP DISPLAY....98
CREATING HELP DISPLAYS, DETAILED REFERENCE....99
THE AUTODEF WINDOW....99
DIRECTIONAL OPTIONS AT THE AUTODEF WINDOW....101
ALTERING THE WINDOW SIZE AND POSITION....102
METHODS OF ALTERING THE WINDOW....102
DIRECTIONAL OPTIONS AT THE WINDOW SIZING WINDOW....104
IDENTIFYING THE FIELD LOCATION....105
DIRECTIONAL OPTIONS AT THE FIELD LOCATION WINDOW....108
IDENTIFYING THE SCREEN....109
DIRECTIONAL OPTIONS AT THE SCREEN IDENTIFICATION WINDOW....110
THE HELP TEXT EDITOR....111
LINE DELETION....113
LINE INSERTION....113
MOVING AND COPYING TEXT LINES....113
TEXT RECORD SPLITS....114
INSERT STRTFLD/STOPFLD....114
DISPLAY ONE OR MORE BLANK LINES - SPACE....114
SKIP THE DISPLAY TO THE NEXT PANEL - EJECT....114
DEFINE THE BEGINNING OF TEXT FOR ONE FIELD - STRTFLD....115
DEFINE THE END OF TEXT FOR ONE FIELD - STOPFLD....115
COPY AND DISPLAY THE TEXT FROM ANOTHER FIELD ID - INCLUDE....115
CREATING COLOR AND HIGHLIGHTING....117
FINDING ATTRIBUTES ON THE SCREEN....119
MARKING AN ATTRIBUTE LOCATION - THE SELECT KEY....119
ENTERING COMMANDS AT THE TEXT EDITOR....120
COMMANDS OF THE EDITOR COMMAND LINE....122
USING THE WHLP DIRECTORY....123
TRANSACTION DIRECTORY COMMANDS....123
FIELD DIRECTORY COMMANDS....124
SYSTEM DIRECTORY COMMANDS....124
MISCELLANEOUS COMMANDS AND FUNCTIONS....125
TRANSACTION KEY RECORD DIRECT MAINTENANCE....125
CROSS DOCUMENT RECORD DIRECT MAINTENANCE....127
RENUMBERING TEXT RECORDS....129
USING THE WORD WRAP FEATURE....129
RESETTING UPPER CASE TRANSLATION....130
CREATING AND USING A TABLE OF CONTENTS....131
SEARCHING FOR TEXT STRINGS IN THE EDITOR....132
IMPORTING HELP TEXT FROM ANOTHER SOURCE....133
METHODS OF CREATING A HELP MENU....135
USING DUMMY TRANSACTIONS....135
HELP ACCESS MODE - USING ON-LINE HELP DISPLAYS....137

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ACTIVATION OF HELP ACCESS MODE....137
RETRIEVING HELP DISPLAYS....138
DIRECTIONAL OPTIONS AT A HELP DISPLAY....140
RETRIEVING A SCREEN-LEVEL HELP DISPLAY....141
RETRIEVING A TRANSACTION-LEVEL HELP DISPLAY....141
CHAINED HELP WINDOWS....142
HELP DISPLAYS WITHIN HELP TEXT (NESTED HELP)....144
EXITING FROM A HELP DISPLAY....145
USING THE CUT/PASTE FEATURE....146

SECTION 8. THE MENU GENERATION FEATURE....149

THE MENU DIRECTORY DISPLAY....150
FIELDS OF THE MENU DIRECTORY DISPLAY....150
ENTERING COMMANDS AT THE MENU DIRECTORY DISPLAY....151
DEFINING AND MAINTAINING MENUS....152
ADDING A NEW MENU....152
CHANGING AN EXISTING MENU....152
DELETING A MENU....152
THE MENU TEXT EDITOR....153
USING THE MENU TEXT EDITOR....153
LINE DELETION....154
LINE INSERTION....154
MOVING AND COPYING TEXT LINES....154
CREATING COLOR AND HIGHLIGHTING....155
THE SELECT WINDOW....157
ENTERING COMMANDS AT THE MENU TEXT EDITOR....158
THE MENU DEFINITION DISPLAY....159
FIELDS OF THE MENU DEFINITION DISPLAY....159
ENTERING COMMANDS AT THE MENU DEFINITION DISPLAY....163
INITIATION AND OPERATION OF MENUS....163
AUTOMATIC INITIATION OF MENUS....163
USING REPLACABLE PARAMETERS IN MENUS....165

SECTION 9. THE MESSAGE BROADCASTING FEATURE....166

THE MESSAGE DIRECTORY DISPLAY....166
FIELDS OF THE MESSAGE DIRECTORY DISPLAY....167
ENTERING COMMANDS AT THE MESSAGE DIRECTORY DISPLAY....167
DEFINING AND MAINTAINING MESSAGES....168
ADDING A NEW MESSAGE....168
CHANGING AN EXISTING MESSAGE....168
DELETING A MESSAGE....168
THE MESSAGE DEFINITION DISPLAY....169
FIELDS OF THE MESSAGE DEFINITION DISPLAY....169
ENTERING COMMANDS AT THE MESSAGE DEFINITION DISPLAY....171
THE MESSAGE TEXT EDITOR....173
USING THE MESSAGE TEXT EDITOR....173
LINE DELETION....173
LINE INSERTION....175
MOVING AND COPYING TEXT LINES....175
CREATING COLOR AND HIGHLIGHTING....176
ENTERING COMMANDS AT THE MESSAGE TEXT EDITOR....177
THE ACTIVE MESSAGE STATUS DISPLAY....178
FIELDS OF THE ACTIVE MESSAGE STATUS DISPLAY....178
ENTERING COMMANDS AT THE ACTIVE MESSAGE STATUS DISPLAY....179
THE TERMINAL STATUS DISPLAY....180
FIELDS OF THE TERMINAL STATUS DISPLAY....180

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE TERMINAL STATUS DISPLAY 180
THE MESSAGE RECEPTION DISPLAY 182
ENTERING COMMANDS AT THE MESSAGE RECEPTION DISPLAY 183
Section 10. INSTALLATION OF THE CICS-WINDOWS PRODUCT 184
CICS REQUIREMENTS (*** MUST READ ***) 184
THE INSTALLATION TAPE 185
KEYWORDS OF THE UNICOM INSTALLATION PROGRAM 185
FORMAT OF THE STSINST KEYWORDS 185
KEYWORD DESCRIPTIONS 185
INSTALLING THE PRODUCT CONTROL PASSWORD 189
DATA FIELDS OF THE PASSWORD CONTROL TABLE 191

SECTION 11. CUSTOMIZATION 192

CONVERTING THE WNDOTBL MACRO TO THE ONLINE VERSION 194
STEPS OF CONVERTING THE WNDOTBL MACRO 194
WNDOTBL MACRO STATEMENT FORMAT 195
INSTALLATION OF THE WNDOTBL MACRO 196
ON-LINE CUSTOMIZATION 197
THE AUXILIARY FUNCTIONS TRANSACTION 199
ENTERING COMMANDS AT THE AUXILIARY FUNCTIONS MENU 199
ACTION BAR FUNCTIONS AT THE AUXILIARY FUNCTIONS MENU 200
THE USER OPTIONS TABLE 201
FIELDS OF THE ON-LINE USER OPTION TABLE 201
ENTERING COMMANDS AT THE USER OPTIONS TABLE 209
THE FILE TABLE 210
FIELDS OF THE FILE TABLE 210
THE USER PROFILE TABLE 212
FIELDS OF THE USER PROFILE TABLE 212
ENTERING COMMANDS AT THE USER PROFILE TABLE 220
THE AUTO-INIT TABLE 221
FIELDS OF THE AUTO-INIT TABLE 221
ENTERING COMMANDS AT THE AUTO-INIT TABLE 226
USING THE AUTO-INIT TABLE TO GENERATE USER PROFILES 227
THE APPLICATION STARTUP TABLE 231
FIELDS OF THE APPLICATION STARTUP TABLE 231
ENTERING COMMANDS AT THE APPLICATION STARTUP TABLE 233
THE TERMINAL EXCLUSION TABLE 234
FIELDS OF THE TERMINAL EXCLUSION TABLE 234
ENTERING COMMANDS AT THE TERMINAL EXCLUSION TABLE 236
THE TRANSACTION EXCLUSION TABLE 237
FIELDS OF THE TRANSACTION EXCLUSION TABLE 238
ENTERING COMMANDS AT THE TRANSACTION EXCLUSION TABLE 238
THE SIGNOFF TRANSACTION TABLE 239
FIELDS OF THE SIGNOFF TRANSACTION TABLE 240
ENTERING COMMANDS AT THE SIGNOFF TRANSACTION TABLE 240
THE SIGNON TRANSACTION TABLE 241
FIELDS OF THE SIGNON TRANSACTION TABLE 242
ENTERING COMMANDS AT THE SIGNON TRANSACTION TABLE 242
THE TERMINAL COMPRESSION TABLE 243
FIELDS OF THE TERMINAL COMPRESSION TABLE 244
ENTERING COMMANDS AT THE TERMINAL COMPRESSION TABLE 244
THE TRANSACTION COMPRESSION TABLE 245
FIELDS OF THE TRANSACTION COMPRESSION TABLE 245
ENTERING COMMANDS AT THE TRANSACTION COMPRESSION TABLE 246
THE SINGLE OCCURRING TRANSACTIONS TABLE 247
FIELDS OF THE SINGLE OCCURRING TRANSACTIONS TABLE 248

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE SINGLE OCCURRING TRANSACTIONS TABLE 248
THE SINGLE OCCURRING PROGRAMS TABLE 249
FIELDS OF THE SINGLE OCCURRING PROGRAMS TABLE 250
ENTERING COMMANDS AT THE SINGLE OCCURRING PROGRAMS TABLE 250
THE STOPPED TRANSACTIONS TABLE 251
FIELDS OF THE STOPPED TRANSACTIONS TABLE 252
ENTERING COMMANDS AT THE STOPPED TRANSACTIONS TABLE 252
THE WINDOW MODE STOPPED TRANSACTIONS TABLE 253
FIELDS OF THE WINDOW MODE STOPPED TRANSACTIONS TABLE 254
ENTERING COMMANDS AT THE WINDOW MODE STOPPED TRANSACTIONS TABLE 254

SECTION 12. SPECIAL CONSIDERATIONS 255

PERFORMANCE TIPS 255
INITIALIZING CICS-WINDOWS IN THE PLT 257
NEWCOPY COMMANDS FOR WINDOWS PROGRAMS 257
PSEUDO TERMINAL ID CONSIDERATIONS 258
AUTO-INITIATED TRANSACTIONS 259
CICS Emergency Restart 259
ICCF REQUIREMENTS AND PECULIARITIES 259
USING THE CURSOR-SELECT KEY AS A HOT KEY 261
SESSION TERMINATION OPTIONS 263
Using the Inactivity Time-Out Feature 267
WLOC Transaction 267
WLSC Transaction 268
Section 13. UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS 269
RUNNING CICS-WINDOWS WITH MRO/ISC 269
MULTIPLE TERMINAL OWNING REGIONS 271
RUNNING WINDOWS WITH THE DOS VSE INTERACTIVE INTERFACE 273
INSTALLING THE MULTIPLE INTERACTIVE INTERFACE FEATURE 274
CHOOSING THE METHOD OF OPERATION 274
INSTALLING THE WNDOVSP PROGRAM 274
ACTIVATING CICS-WINDOWS FROM A SELECTION PANEL 275
CONSIDERATIONS FOR USING ICCF AND CICS-WINDOWS IN VSE 276
OVERHEAD CONSIDERATIONS WITH MULTIPLE INTERACTIVE INTERFACE SESSIONS 277

SECTION 14. SPECIAL-PURPOSE COMMANDS 278

COMMAND SYNTAX AND USAGE 278
BACKOUT 278
CPROFF 279
CPRON 279
DEBUG 279
INIT 279
START 282
STOP 282
TEMP 283

SECTION 15. USER EXITS AND PROGRAM INTERFACES 284

WRITING A USER EXIT FOR CICS-WINDOWS 284
TRANSACTION ATTACH EXIT 285
PSEUDO TERMINAL ID GENERATION EXIT 287
INSTALLING THE USER EXIT PROGRAM 289
THE MENU GENERATION FEATURE USER EXIT 290
INSTALLING THE MENU GENERATION FEATURE USER EXIT 293
PROGRAM INTERFACE ROUTINES 293
THE APPLICATION PROGRAM INTERFACE 294
FUNCTIONS AVAILABLE 294

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

LINKING TO CICS-WINDOWS 296
PASSING COMMANDS 296
INVOKING APPLICATION PROGRAMS 300
EXAMPLES OF APPLICATION INVOCATION 300
TESTING SUCCESSFUL EXECUTION 302
ACTIVATING AND DEACTIVATING CICS-WINDOWS FROM A USER PROGRAM 302
LINKING FROM A MACRO LEVEL PROGRAM 304
Macro Level Assembler Examples 305
LINKING FROM A COMMAND LEVEL PROGRAM 306
Command Level COBOL Example 306
BYPASSING THE CICS-WINDOWS USER CONFIGURATION DISPLAY 307
BUILT-IN FUNCTIONS 308
OBTAINING THE BUILT-IN FUNCTION ENTRYPOINT 308
THE BUILT-IN FUNCTION ENTRYPOINT LIST 310
LINKAGE CONVENTIONS 310
THE BUILT-IN FUNCTIONS, CODING DETAILS AND EXAMPLES 312
USES FOR THE BUILT-IN FUNCTIONS 315
DEACTIVATING CICS-WINDOWS WHEN A NODE ERROR OCCURS 315
CODING THE DFHZNEP INTERFACE 316
DEACTIVATING PRIOR TO AUTOMATIC LOG-OFF OR SIGN-OFF 316
OBTAINING THE REAL TERMINAL ID FROM A PSEUDO ID 318
CODING THE ACF2 TERMINAL ID RETRIEVAL EXIT 318
ACCESSING OR ALTERING THE ACTIVE USER TABLE ENTRY 318
THE ACTIVE USER TABLE 319
CODING THE ACTIVE USER TABLE ACCESS INTERFACE 319
ALTERING THE ACTIVE USER TABLE ENTRY 320
CHANGING THE TRANSACTION CODE IN A RECURRING SESSION 321
CODING THE AUTO-START TRANSACTION-CHANGE INTERFACE 321
SAMPLE PROGRAMS ON THE INSTALLATION TAPE 323
PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3 325
ACCESSING THE TCA AND CSA 325

SECTION 16. MESSAGES 326

AUXILIARY FUNCTIONS MESSAGES "WA" PREFIX 326
MESSAGE BROADCASTING MESSAGES "WB" PREFIX 332
DEMONSTRATION PROGRAM MESSAGES, “WD” Prefix 336
WNDOHELP MESSAGES "WH" PREFIX 340
WNDOINIT MESSAGES "WI" PREFIX 354
MENU GENERATION MESSAGES "WM" PREFIX 355
WNDOMRO MESSAGES "WMR" PREFIX 357
TEXT EDITOR MESSAGES "WN" PREx" 358
CICS-WINDOWS MESSAGES "WNDO" PREFIX 359

APPENDIX A - SUMMARY OF CICS-WINDOWS COMMANDS 373

APPENDIX B - SYSTEM REQUIREMENTS 377

APPENDIX C – INSTALLATION INSTALLATION OF THE CICS-WINDOWS PRODUCT 379

CICS REQUIREMENTS (*** MUST READ ***) 379
THE INSTALLATION TAPE 380
DOS/VSE INSTALLATION 380
MVS INSTALLATION 382
DEFINING THE DFHZNEP INTERFACE 387
ADDITIONAL INSTALLATION NOTES, MVS AND DOS/VSE 387
SUMMARY 387

APPENDIX D - REPORTING PROBLEMS 403

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

GLOSSARY OF TERMS **404**
INDEX **407**

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PREFACE

This book is for anyone who is responsible for the installation, administration, and proliferation of CICS-WINDOWS at the user site. Although this book can and should be used by the average user of the CICS-WINDOWS product, the CICS-WINDOWS Terminal Users Guide may be more appropriate for the new users who desired to quickly learn how to use the CICS-WINDOWS product. If you do not have CICS-WINDOWS Terminal User Guides, they are available from Unicom Systems by calling (818) 838-0606. They may be copied and distributed by licensed users of CICS-WINDOWS for internal use.

This book is comprised of the following sections:

Section 1. Introduction **Page 16**

This section describes CICS-WINDOWS in general terms as to what it can do in your working environment and a few of the ways that it can be used.

Section 2. Using Applications with CICS-WINDOWS **Page 19**

This section provides a step-by-step example of a typical set of user applications using the CICS-WINDOWS product. The user is taken through CICS sign-on, CICS-WINDOWS initialization, starting applications, and moving between applications (toggling). A discussion of the various commands and functions available to control the product can be found in this section.

Section 3. Using the Windowing Features **Page 23**

This section describes the windowing features of CICS-WINDOWS is detail. Window configurations and initialization are discussed. Instructions for entering and exiting window mode, starting application transactions from within windows, changing the windows size, and moving between windows are provided.

Section 4. Examples of Use **Page 50**

This section provides typical examples of situations that are common to many environments. It illustrates where CICS-WINDOWS is used to increase productivity and enhance the normal application process.

Section 5. Operation **Page 52**

This section is designed to help you learn CICS-WINDOWS quickly. It describes how to get started and define any keys you will use. It gives an explanation of the commands that are commonly used in operation of CICS-WINDOWS.

Section 6. Using the Cut and Paste Feature **Page 69**

A high productivity feature of CICS-WINDOWS is the cut and paste feature. This allows copying fields, lines, or an entire screen. A detailed description of the use of this feature is provided in this section.

Section 7. Help Windows Feature **Page 88**

An optional feature of CICS-WINDOWS is a facility which enables users to create online help displays for any CICS transaction. Help displays can be created to display either in full-screen mode or in a pop-up window on the screen. Help can be accessed at the transaction level, the screen level, and/or the field level. This section describes the use of the help feature.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 8. Menu Generation Feature **Page 142**

Another optional feature of CICS-WINDOWS, Menu Generation, is a facility that enables users to create menus to be used to increase the performance and operation of existing online applications. A menu is configured such that it invokes transactions, programs, or other menus. This section details the generation and use of menus.

Section 9. Message Broadcasting Feature **Page 158**

Messages can be sent from one terminal operator to another or to an entire group of operators using the optional feature of CICS-WINDOWS known as Message Broadcasting. This section describes how this feature is used.

Section 10. Installation **Page 174**

All information required to install the CICS-WINDOWS product and make it operational can be found in this section.

Section 11. Customization **Page 186**

Options are available in CICS-WINDOWS to customize it to the specific needs of your environment. This section describes the options available and how to set them.

Section 12. Special Considerations **Page 248**

Technical notices and considerations that the systems programmers should be aware of are found in this section.

Section 13. Unique Environments and Special Situations **Page 262**

This sections deals with any environmental situations which are non-standard. In particular, the MRO/ISC environment and the DOS/VSE Interactive Interface are discussed.

Section 14. Special Purpose Commands **Page 272**

This section describes the available system commands that can be used to control the CICS-WINDOWS environment.

Section 15. User Exits and Program Interfaces **Page 278**

This section describes the various user exits and program interface routines that can be used to enhance or control the CICS-WINDOWS environment.

Section 16. Messages **Page 312**

When various conditions are encountered, CICS-WINDOWS displays a standard message to help the operator understand the situation and circumstances. This sections provides explanations for the standard messages that CICS-WINDOWS may display. These conditions may be due either to terminal operator error or system problems. Messages are listed in sequence by the message number followed by a suggested action.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Appendices **Page 358**

Appendices are provided to valuable information including:

- Summary of CICS-WINDOWS Commands
- Description of system requirements, storage usage and technical data
- Installation documentation that may be printed during installation
- Procedures to use if problems are encountered

Glossary of Terms **Page 390**

A glossary of terms is provided with explanations for some of the key words and phrases used throughout this book.

Index **Page 393**

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 1. INTRODUCTION

CICS-WINDOWS -- A powerful productivity tool.

CICS-WINDOWS is one of the most significant software tools for improving productivity of online systems. It goes beyond the definition of a multiple sessions manager; it is a true multiple windows manager for CICS. It allows you to conduct up to nine concurrent CICS sessions and open as many windows simultaneously on one screen. Each window is fully functional and compatible with every type of CICS application. The windows may be thought of as being mini-terminals in that all of the windows have the identical capabilities to the full screen. There is no need for any hardware changes. There is no need to modify any existing software. You will not notice any degradation to the system or the user response time. CICS-WINDOWS is easy to learn and easy to use.

With **CICS-WINDOWS** it is now possible to:

- Have window facilities on every mainframe terminal without any changes to hardware or software.
- Have the information from several transactions available on a single screen.
- Easily transfer information from one transaction to another.
- Work with more than one transaction on the same screen.
- Expand terminal capabilities by having up to nine virtual terminals on each physical terminal.
- Eliminate the time consuming requirement of logging on and off when switching between applications.

You may have up to nine different window configurations opened simultaneously on the same physical screen. The effect is similar to having nine separate physical terminals. You can instantly toggle (switch) between window mode and full screen using a single keystroke.

CICS-WINDOWS increases productivity by having needed information simultaneously available on the same screen. You can have on your screen, all at the same time, application screens, help windows, menus, prior transactions, editors, notepads, and anything else that you may want to use.

CICS-WINDOWS is very user friendly and easy to use. It operates with or without PF and/or PA keys. You have complete control over whichever application is currently selected, and you may easily move from one window to another, using only one keystroke.

In addition to the various user functions, CICS-WINDOWS provides a performance enhancing feature to all CICS online users. CICS-WINDOWS provides data compression to optimizes all terminal datastreams, whether they are using CICS-WINDOWS or not. This data compression feature can provide as much as 60 percent improvement in the throughput of CICS transactions.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Terminology

First, a few words about terminology as it applies to the CICS-WINDOWS product. Throughout this document, we refer to terms like “sessions”, “windows”, “virtual terminals”, etc., which may be unfamiliar. Following the appendices is a *Glossary of Terms*, which describes most of the words used, but for now, here are the most frequently-used expressions and their meaning:

APPLICATION

This refers to a program or transaction which runs in your CICS environment. An application often consists of several programs or transactions. Thus, we might refer to Data Entry as an application, when several transactions are involved to perform all Data Entry functions.

FULL SCREEN

This term is used to designate the normal way that transactions are run. That is, the screen is not divided up into multiple parts, or windows.

FULL SCREEN MODE

The normal mode of operation, where there are no windows on the screen.

HOT KEY

A “hot” key is the term used to indicate a PF or PA key which, when pressed, will activate some CICS-WINDOWS function rather than perform its normal function in CICS. A “hot” key always overrides the application program running in a virtual terminal. In CICS-WINDOWS various keys can be assigned to be “hot” keys. They are: the Toggle Forward key, the Toggle Backward key, Direct Session keys and the Window key. You can use a minimum of one “hot” key, and more if enough PF/PA keys are available.

SESSION

This term is used interchangeably with VIRTUAL TERMINAL. In the literal sense, it refers to a virtual terminal when an application program has been started in it, as opposed to one where no transaction is active.

SPLIT SCREEN

This term is used interchangeably with WINDOWS. It refers to the ability of the CICS-WINDOWS user to divide the screen into two or more parts, each part (window) displaying data from one SESSION.

TASK

The term TASK can usually be used interchangeably with TRANSACTION. Strictly speaking, a task consists of everything that happens in the computer from the time that a transaction code is entered until a display is received on the screen. There could be several programs involved in a single task.

TRANSACTION

A transaction is a function performed on a terminal which consists of entering data, pressing the Enter key or some program function key and receiving a display on the screen. A transaction is initiated by entering a 1 to 4 character code, known as the transaction code, or sometimes by simply pressing one of the PF or PA keys. Generally speaking, there is a transaction code associated with each program.

VIRTUAL TERMINAL

A virtual terminal is a logical extension of a physical terminal. The term refers to the ability of CICS-WINDOWS to redefine a physical terminal as if it were two or more (up to nine) physical terminals. Each virtual terminal can do anything that the physical terminal can do. You move from virtual terminal to virtual terminal by pressing a “toggle” key. Thus a physical terminal defined by CICS-WINDOWS as, for example, four virtual terminals would be like having four physical terminals setting on your desk.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WINDOW

A window is a portion of the physical screen display. There can be from 2 to 9 windows simultaneously displayed on the screen. Windows may vary in size and shape. They are bordered with a row of underscores at the top and bottom, and a column of vertical bars on the left or right side. A window can be empty (clear) or it can contain (display) data from the SESSION which is active in a VIRTUAL TERMINAL.

WINDOW MODE

The method of operation where the screen is divided into multiple parts (windows). A transaction is initiated in a window by designating an ACTIVE window and entering the transaction code as if it were a full screen.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Product Definition

CICS-WINDOWS is a software product running in CICS that extends the capability of a physical terminal to act as if it were up to nine physical terminals. Each additional virtual terminal, or session, can be accessed via a Hot Key, or by various other methods.

In addition, CICS-WINDOWS will bring some or all sessions together on the physical screen in parts, or windows, each of which can be operated just like a full-screen application.

The CICS-WINDOWS Demonstration Program

Contained on the installation tape is an on-line demonstration program, the name of which is WNDODEMO. This program will exercise most of the CICS-WINDOWS features in a "hands-free" demonstration. It is a good idea to run the demonstration immediately after installation, for two reasons:

The first thing WNDODEMO does is verify the installation to ensure everything was installed correctly. Any installation errors will be reported.

The subsequent demonstration will illustrate many of the things that can be done with the product, and may give you ideas for your own installation.

RUNNING THE DEMONSTRATION PROGRAM

To activate the demo, simply enter

WNDO,ON,DEMO

from a clear screen. If desired, entering the WDMO transaction code will produce the same results.

The demo should be activated at a color terminal that supports extended attributes, otherwise you will not be able to see the various color and highlighting effects of the windows.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 2. USING APPLICATIONS WITH CICS-WINDOWS ACTIVE

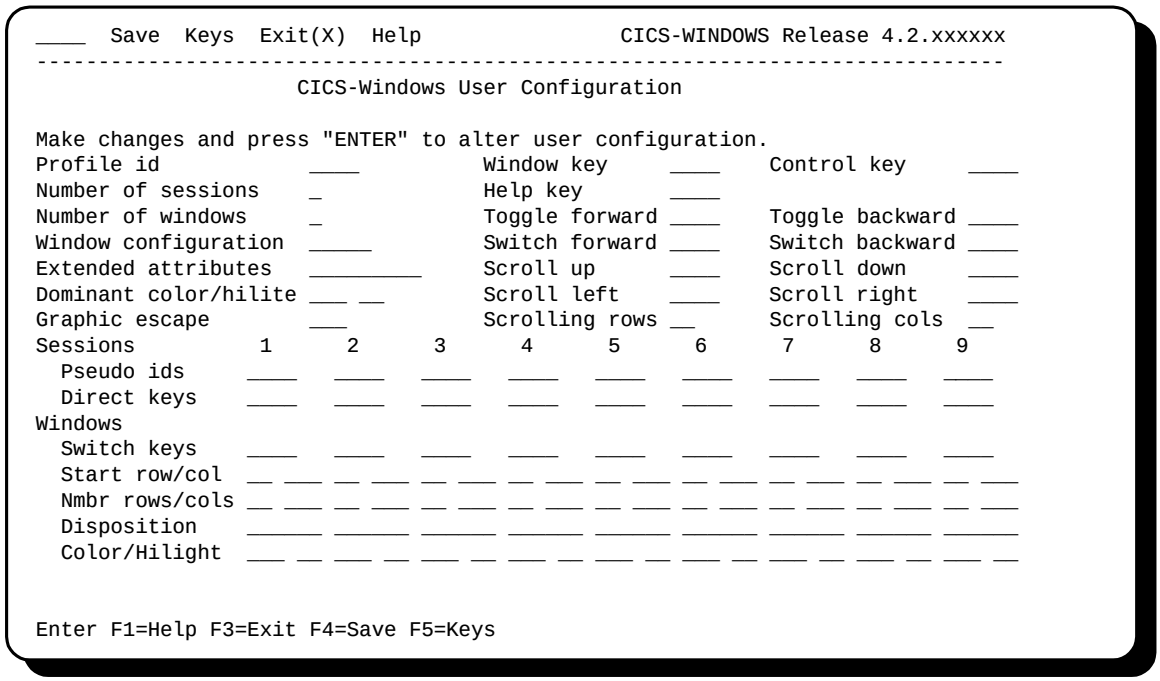
This section will discuss how to operate applications while using CICS-WINDOWS. Examples will be given to aid the user.

TUTORIAL

First, sign-on to CICS using your normal operator name and password. Signing on prior to activating CICS-WINDOWS will allow the same security to be established for all virtual terminals.

Now initiate CICS-WINDOWS with the WNDO,ON command.

Upon completion, the User Configuration display will appear with the configuration that has been defined for your terminal.


____ Save Keys Exit(X) Help CICS-WINDOWS Release 4.2.xxxxxx
CICS-Windows User Configuration
Make changes and press "ENTER" to alter user configuration.
Profile id ____ Window key ____ Control key ____
Number of sessions _ Help key ____
Number of windows _ Toggle forward ____ Toggle backward ____
Window configuration _____ Switch forward ____ Switch backward ____
Extended attributes _____ Scroll up ____ Scroll down ____
Dominant color/hilite ___ __ Scroll left ____ Scroll right ____
Graphic escape ___ Scrolling rows __ Scrolling cols __
Sessions 1 2 3 4 5 6 7 8 9
Pseudo ids
Direct keys
Windows
Switch keys
Start row/col
Nmbr rows/cols
Disposition
Color/Hilight
Enter F1=Help F3=Exit F4=Save F5=Keys

Memorize or jot down the keys designated for the Toggle Forward, Help and Window keys. These are the most used. Note that if a Help key is not specified, PF1 is the Help key in all CICS-WINDOWS screens. Remember that any time you need to return to this screen to display your keys you may simply clear the screen, key the transaction, WNDO, and press Enter.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

At this point you may press PF3 to exit the display and proceed with whatever application is desired. Follow your normal procedures. For example, the following transaction may be initiated:

```
LKUP,A*

 ABERNATHY, DAWN M.        06378   KANSAS CITY, MO
 ABBOT, KARRON L.          00423   OKLAHOMA CITY, OK
 ABRAHAM, AARON            99332   SEATTLE, WA
 ANDERSON, BOBBIE          26932   NASHVILLE, TN
 ANDERSON, J. L.           00412   NEW YORK, NY
 ARBLE, MARY A.            03500   SAN DIEGO, CA
 ASHLEY, WILLIAM           63860   ARDMORE, OK
 ASNER, JOHN A.            12654   DALLAS, TX
 BARKLEY, RONALD           02564   CHICAGO, IL
 BRADLEY, JACK W.          00163   FT. WORTH, TX
 BRADLEY, JOHN S.          10456   MEMPHIS, TN
 BREKHAN, ALLEN R.         03664   LOS ANGELOS, CA
 CACHET, GEORGE C.         44512   DETROIT, MI
 COUCH, WILLIAM E.         09432   BETHANY, OK
 CREED, ROBERT A.          00087   WASHINGTON, DC
 DANELY, JAMES L.          74811   OKLAHOMA CITY, OK
 DOVER, EARL III           36029   LUNAR SURFACE
 DUGAN, MAX L.             98473   NEW YORK, NY
 DUNLAP, SAMANTHA          34801   MELBOURNE, AUST
 DUNN, RALPH K.            09358   SAN DIEGO, CA

PF3=EXIT   PF7=BACKWARD   PF8=FORWARD
```

At any time you may press the "Toggle forward" key. Your current application will be saved and CICS-WINDOWS will proceed to the next virtual terminal. It will initially be a blank screen.

You may now initiate another CICS application on this virtual terminal.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
                  NAME AND ADDRESS MAINTENANCE
    COMPANY NAME ____________________________________________________________
    COMPANY ADDRESS _________________________________________________________
    CITY ___________________________________ STATE ___________ ZIP _________
    BANK NAME _______________________________________________________________
    TRANSIT NO. ____________________ ACCOUNT NUMBER _____________________


    INSURED'S NAME __________________________________________________________
    ADDRESS _________________________________________________________________
    CITY ___________________________________ STATE ___________ ZIP _________


          KEY ALL REQUIRED FIELDS AND PRESS ENTER TO ADD RECORD
```

Each time the Toggle Forward key is pressed, CICS-WINDOWS moves to the next virtual terminal.

If you are positioned at the last virtual terminal (equal to the number of sessions available on this terminal) and press the Toggle Forward key, you will be positioned back to virtual terminal number one.

```
LKUP,A*

 ABERNATHY, DAWN M.        06378   KANSAS CITY, MO
 ABBOT, KARRON L.          00423   OKLAHOMA CITY, OK
 ABRAHAM, AARON            99332   SEATTLE, WA
 ANDERSON, BOBBIE          26932   NASHVILLE, TN
 ANDERSON, J. L.           00412   NEW YORK, NY
 ARBLE, MARY A.            03500   SAN DIEGO, CA
 ASHLEY, WILLIAM           63860   ARDMORE, OK
 ASNER, JOHN A.            12654   DALLAS, TX
 BARKLEY, RONALD           02564   CHICAGO, IL
 BRADLEY, JACK W.          00163   FT. WORTH, TX
 BRADLEY, JOHN S.          10456   MEMPHIS, TN
 BREKHAN, ALLEN R.         03664   LOS ANGELOS, CA
 CACHET, GEORGE C.         44512   DETROIT, MI
 COUCH, WILLIAM E.         09432   BETHANY, OK
 CREED, ROBERT A.          00087   WASHINGTON, DC
 DANELY, JAMES L.          74811   OKLAHOMA CITY, OK
 DOVER, EARL III           36029   LUNAR SURFACE
 DUGAN, MAX L.             98473   NEW YORK, NY
 DUNLAP, SAMANTHA          34801   MELBOURNE, AUST
 DUNN, RALPH K.            09358   SAN DIEGO, CA

PF3=EXIT   PF7=BACKWARD   PF8=FORWARD
```

Now, each time you press the Toggle Forward key, control will proceed to the next sequential virtual terminal and re-display it exactly as it was left.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If Direct Session keys or the Toggle Backward key have been defined for this terminal, there are additional methods of moving from one virtual terminal to another.

TOGGLE BACKWARD KEY OPERATION

The Toggle Backward key works exactly like the Toggle Forward key in reverse. Each time it is pressed, control moves to the previous sequential terminal until virtual terminal 1 is reached. When pressed again, control moves to the last virtual terminal.

For example, assume four virtual terminals are established and virtual terminal 4 is the current active session. Pressing the backward-toggle key four times would cause a switch from terminal 4 to 3, terminal 3 to 2, terminal 2 to 1, then from terminal 1 back to 4 again.

The User Configuration display will show if Direct Session or Toggle Backward keys are available on your terminal. If not, you may alter the appropriate field(s) of the User Configuration display for the keys that you wish (for more information, see THE USER CONFIGURATION DISPLAY in section 02 - OPERATION). Your MIS contact person or system programmer should be able to make these keys automatically available upon WINDOWS initialization if you want to use them (see THE USER PROFILE TABLE in section 11 - CUSTOMIZATION).

DIRECT SESSION KEY OPERATION

Direct Session keys provide a unique PF or PA key for one or all of the virtual terminals that are in use. Pressing one of these keys will transfer control directly to the corresponding virtual terminal without the necessity of moving sequentially through all sessions.

To illustrate, suppose you have four virtual terminals established, PF21 is the Direct Key for session 1, PF22 for session 2, PF23 for session 3 and PF24 for session 4. If you are currently operating a transaction in session 1, pressing PF24 would immediately switch to session 4, without toggling through sessions 2 and 3. Pressing PF22 would switch to session 2, etc.

Direct session keys may also be used in Window Mode, to move the requested session directly to either the Dominant window or to window 1, depending on the type of window configuration in use (Popup, Variable or Standard).

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 3. USING THE WINDOWING FEATURE

This section describes the procedures used to window data in various sessions together on the same screen.

ENTERING WINDOW MODE

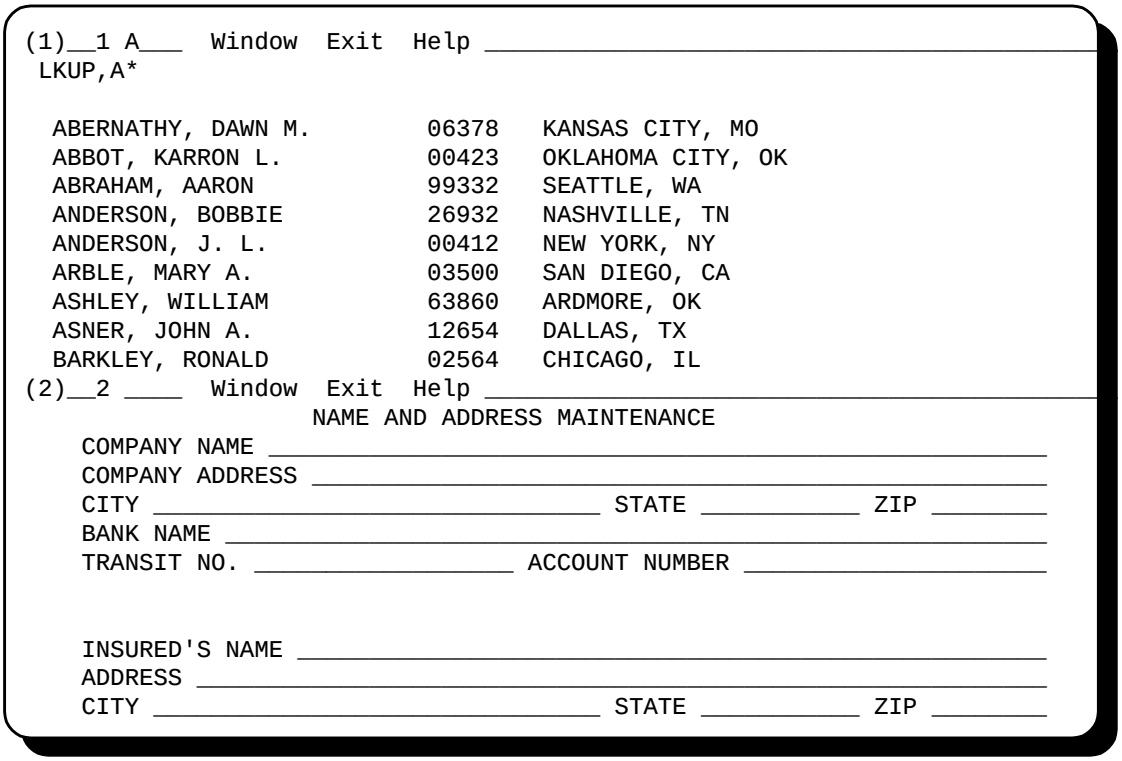
At any point during CICS-WINDOWS operation you may press the Window key, which was established at initialization. (Pressing the Window key once enters window mode, pressing it again will exit window mode). Upon entering window mode, the screen display will split into multiple sections, each section separated by a dotted line. The sections are called "windows" and the dotted lines are called "window borders". Part of each session display will be available in each window.

The initial size and position of the windows, as well as the type of window mode, may be altered by customization options.

OPERATION WITH WINDOWING

Upon pressing the Window key the screen will split into the window configuration. Current applications of each virtual terminal will be displayed in their respective windows.

In this example, we have a name and address lookup transaction in window 1 (session 1) and a name and address maintenance application in window 2 (session 2).


(1)__1 A___ Window Exit Help
LKUP,A*
ABERNATHY, DAWN M. 06378 KANSAS CITY, MO
ABBOT, KARRON L. 00423 OKLAHOMA CITY, OK
ABRAHAM, AARON 99332 SEATTLE, WA
ANDERSON, BOBBIE 26932 NASHVILLE, TN
ANDERSON, J. L. 00412 NEW YORK, NY
ARBLE, MARY A. 03500 SAN DIEGO, CA
ASHLEY, WILLIAM 63860 ARDMORE, OK
ASNER, JOHN A. 12654 DALLAS, TX
BARKLEY, RONALD 02564 CHICAGO, IL
(2)__2 ____ Window Exit Help
NAME AND ADDRESS MAINTENANCE
COMPANY NAME
COMPANY ADDRESS
CITY STATE ZIP
BANK NAME
TRANSIT NO. ACCOUNT NUMBER
INSURED'S NAME
ADDRESS
CITY STATE ZIP

Each window area can be thought of as a mini-terminal. The top line of each window is the Window Control Line. It looks like this for window 1, virtual terminal 1:

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Each window area can be thought of as a mini-terminal. The top line of each window is the Window Control Line. It looks like this for window 1, virtual terminal 1:

Imbedded in the window control line is a number enclosed in parentheses. This is the window number. There is also a number which could be from 1 to 9 followed by four spaces. This is the virtual terminal number. It is possible for the virtual terminal number to be different from the window number, after toggling while in window mode or if there are more virtual terminals than there are windows.

Following the virtual terminal number and the four spaces are three words where the cursor will stop if the TAB key is pressed. These are pull-down menus, activated by tabbing to the word and pressing Enter. First is the Window control menu, which provides methods of sizing and moving the window. Second is the Exit menu, which provides a method of returning to full-screen mode. Finally there is the Help menu, providing various help topics for operation while in window mode.

None of the pull-down menus have to be used. They are provided as an alternative to PF keys and commands which will perform the same functions.

The four spaces following the virtual terminal number is the window command input area for each window. The command input area is used for entering window commands to perform functions such as:

- Scrolling the application within the window.
- Designating the active window for active transaction input.
- Clearing a window.
- Expanding a window into full-screen mode.
- Altering the size of the window.
- Switching to another window for active transaction input.

These windowing functions are described in the following topics.

SCROLLING COMMANDS

When data is displayed in a window, you of course can not see all of the data that you would see if you were in full-screen mode. The 3270 hardware does not allow characters to be compressed to a smaller font size as do many micro-computers.

There are two ways to see more of the data in a window:

1) Making the window bigger. This is possible and the method for doing this is discussed later in this section under WINDOW ADJUSTMENT COMMANDS.
2) Shift, or "scroll" the data up, down, right or left in the window to see the part that is hidden. This is called "scrolling", and the commands for it are as follows:

Four scrolling commands are used to logically move the window over the desired application area. There are four directional codes: "U" (up), "D" (down), "R" (right) and "L" (left). Each directional code is followed by a one or two digit number designating the number of rows or columns to move.

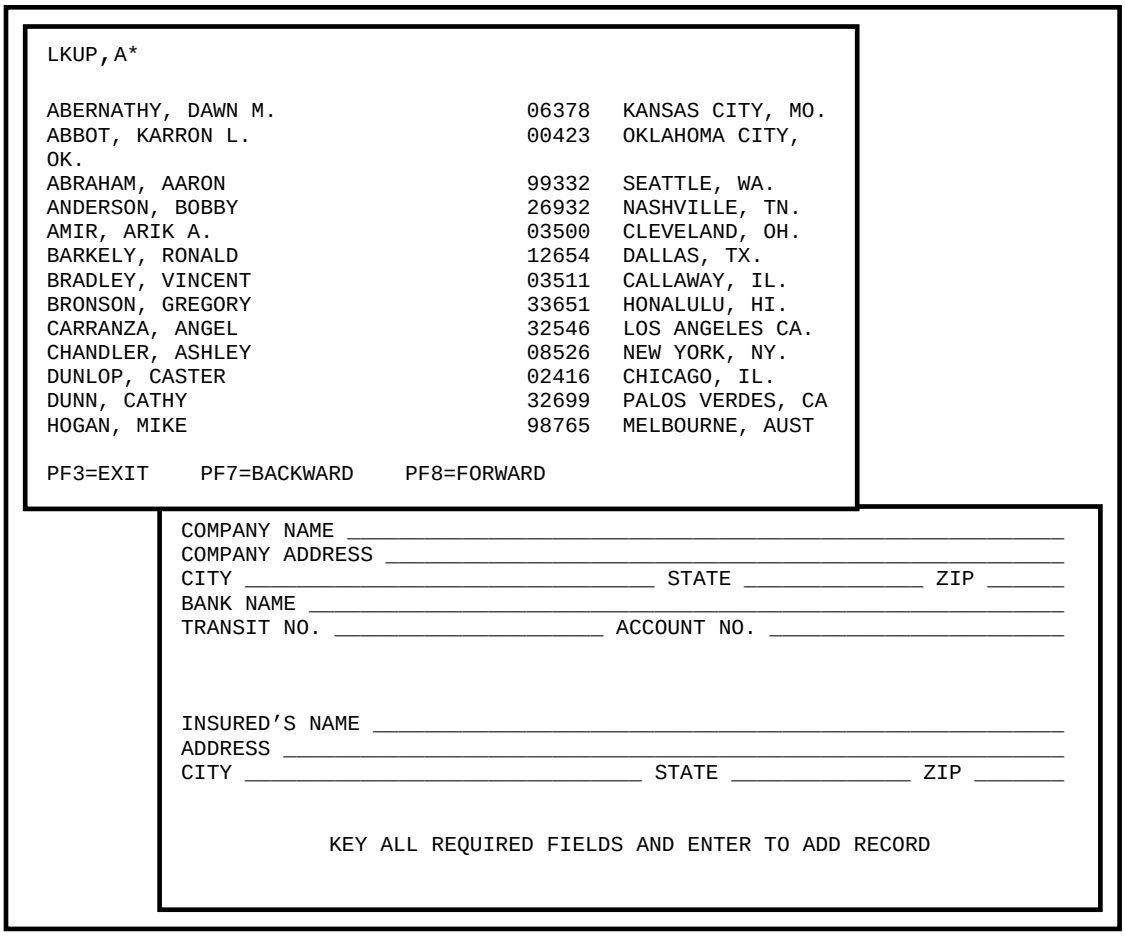
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

When you enter a scrolling command, you are telling CICS-WINDOWS to "move" the window up, down, left or right while the data behind the window remains stationary.

If the command is entered without a number following it, the window will be moved across the data for the number of columns or rows that is specified in the Scrolling Columns or Scrolling Rows field on the User Configuration Display. A scrolling command may be entered in one or more windows at a time.

Here is an illustration of scrolling using a two-window Configuration:

Upon first pressing the window key, the screen is displayed as illustrated.


LKUP,A*
ABERNATHY, DAWN M. 06378 KANSAS CITY, MO.
ABBOT, KARRON L. 00423 OKLAHOMA CITY,
OK.
ABRAHAM, AARON 99332 SEATTLE, WA.
ANDERSON, BOBBY 26932 NASHVILLE, TN.
AMIR, ARIK A. 03500 CLEVELAND, OH.
BARKELY, RONALD 12654 DALLAS, TX.
BRADLEY, VINCENT 03511 CALLAWAY, IL.
BRONSON, GREGORY 33651 HONALULU, HI.
CARRANZA, ANGEL 32546 LOS ANGELES CA.
CHANDLER, ASHLEY 08526 NEW YORK, NY.
DUNLOP, CASTER 02416 CHICAGO, IL.
DUNN, CATHY 32699 PALOS VERDES, CA
HOGAN, MIKE 98765 MELBOURNE, AUST
PF3=EXIT PF7=BACKWARD PF8=FORWARD
COMPANY NAME
COMPANY ADDRESS
CITY STATE ZIP
BANK NAME
TRANSIT NO. ACCOUNT NO.
INSURED'S NAME
ADDRESS
CITY STATE ZIP
KEY ALL REQUIRED FIELDS AND ENTER TO ADD RECORD

Note that the scrolling commands remain in place, once entered. This allows you to continue pressing ENTER to repeat the command. Note also that the 'R' command that was entered in window 1 (with no number following) defaulted to 10 columns

When a window reaches its data border (left, right, top or bottom), it will not scroll any further. Although subsequent commands in the same direction may be entered, a message will be displayed to let you know that you have reached the edge of the data document

WINDOW SCROLLING KEY OPERATION

If Window Scrolling Keys have been assigned to this terminal (see THE USER CONFIGURATION DISPLAY in section 02 - OPERATION) you may simply press the assigned PF or PA key which corre-

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

sponds to the UP, DOWN, LEFT and RIGHT scrolling commands instead of entering the commands in the window command area

When the P or DOWN Scrolling key is pressed, the window will be panned up or down the number of rows specified in the SCROLLING ROW operand of the User Configuration Display.

When the RIGHT or LEFT Scrolling key is pressed, the window will be panned right or left the number of columns specified in the SCROLLING COLUMNS operand of the User Configuration Display.

SWITCHING TO FULL-SCREEN MODE

To expand a window to full-screen mode, enter an "X" in the window control line of the window to be expanded and press ENTER. The application displayed in that window will then expand into full-screen mode.

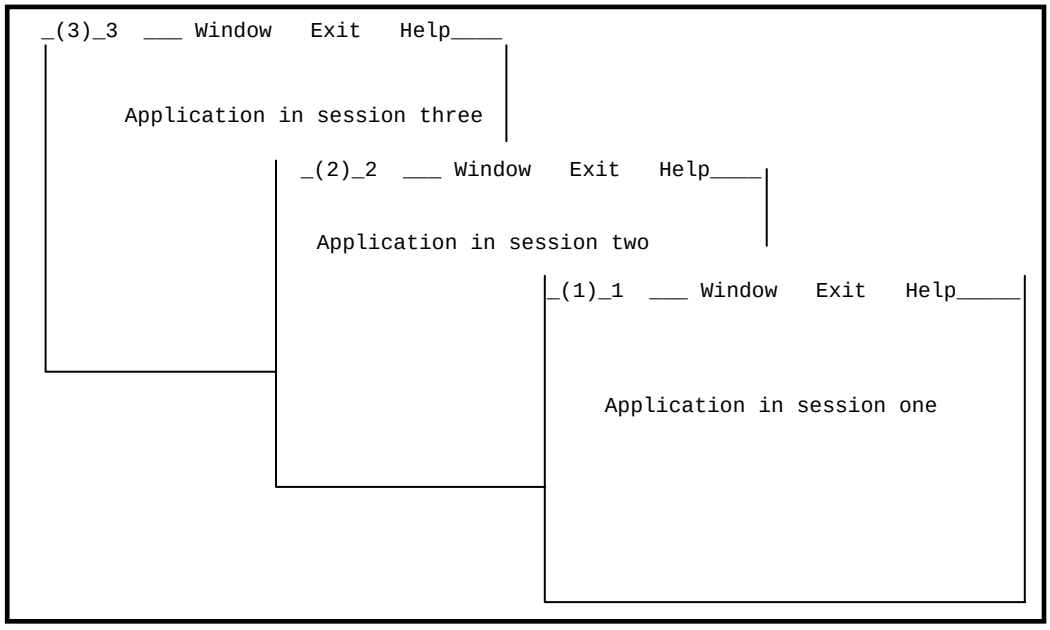
You may also switch to full-screen mode by pressing the "window" key thus exiting window mode. This will expand the session from which window mode was last entered.

TOGGLING WHILE IN WINDOW MODE

If the "Toggle Forward" key is pressed while in window mode, all virtual terminals rotate counter-clockwise one full window. If there are more virtual terminals than there are windows, the next sequential virtual terminal is displayed where the highest numbered virtual terminal had been, and the lowest numbered virtual terminal disappears from view. The virtual terminal numbers on the window command lines will change while the window numbers remain the same.

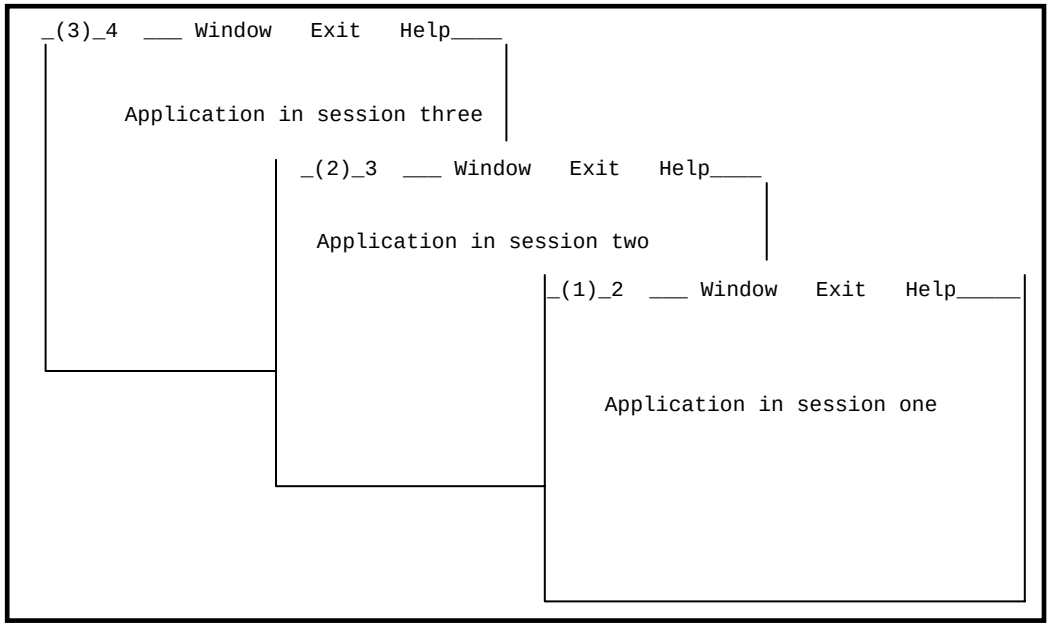
Here is an example of toggling while in window mode:

Assume three windows and four virtual terminals.

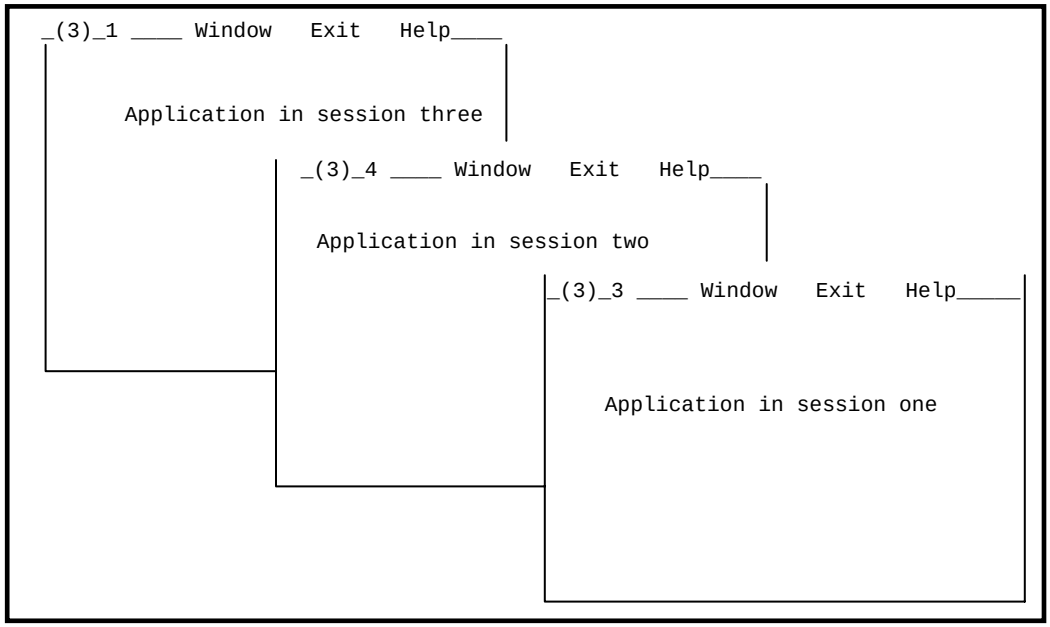

_(3)_3 ___ Window Exit Help____
Application in session three
_(2)_2 ___ Window Exit Help____
Application in session two
_(1)_1 ___ Window Exit Help_____
Application in session one

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

After pressing the "toggle" key ...


_(3)_4 ___ Window Exit Help____
Application in session three
_(2)_3 ___ Window Exit Help____
Application in session two
_(1)_2 ___ Window Exit Help_____
Application in session one

After pressing the "toggle" key again ...


_(3)_1 ____ Window Exit Help____
Application in session three
_(3)_4 ____ Window Exit Help____
Application in session two
_(3)_3 ____ Window Exit Help_____
Application in session one

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WINDOW ADJUSTMENT COMMANDS

Window Adjustment Commands allow you to change the size of any or all windows and, if using Standard Window mode, will establish a DOMINANT WINDOW.

To use Window Adjustment Commands, press the "window" key to enter window mode. It does not matter Whether you have transaction data displayed in the window or not.

Key one of the following commands in the Window Command area above the window to be adjusted. (If using Variable or Popup window modes, only one window can be adjusted at a time).

WRxx - Window right xx columns
WLxx - Window left xx columns
WDxx - Window down xx rows
WUxx - Window up xx rows

The action taken varies depending on the current position of the window, as follows: Note that since Popup Windows are not tied to the edge of the screen, the following commands will move the entire window in the specified direction.

WRxx - If the right-most boundary of the window is NOT the right edge of the terminal screen, this command will move the right boundary bar to the right, thus widening the window.

If the right-most boundary is at the edge of the screen, the left boundary bar is moved to the right, thus narrowing the window.

WLxx - If the left-most boundary of the window is not at the left edge of the terminal screen, the left boundary-bar is moved to the left, thus widening the window

If it is at the edge of the screen, the right boundary bar is moved to the left, thus narrowing the window.

WDxx - If the bottom boundary of the window is not at the bottom edge of the terminal screen, the lower boundary bar is moved down, thus lengthening the window.

If it is at the bottom of the screen, the top boundary bar is moved down, thus shortening the window.

WUxx - If the top boundary of the window is not at the top edge of the terminal screen, the upper boundary bar is moved up, thus heightening the window.

If it is at the top of the screen, the lower boundary bar is moved up, thus shortening the window.

[Note]: For POPUP windows, the above Window Adjustment Commands move the entire window in the specified direction . To alter the size of the window, the Window Sizing Functions must be used. Two methods of window sizing are provided. One method is to use the Window Action Bar pull-down menu. (this is explained in more detail later in this section under ACTION BAR FUNCTIONS OF POPUP WINDOWS.) The second method is to use the following commands:

WWxx - Window wider xx columns
WNxx - Window narrower xx columns
WTxx - Window taller xx rows
WHxx - Window shorter xx rows

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

These commands will move either the right or the bottom window boundary bar, and operate as follows:

WWxx - This command will move the right boundary bar to the right, thus widening the window.
WNxx - This command will move the right boundary bar to the left, thus narrowing the window.
WTxx - This command will move the bottom boundary bar downwards, thus lengthening the window.
WHxx - This command will move the bottom boundary bar upwards, thus shortening the window.

WINDOW ADJUSTMENT RULES:

1) A window can not be taller than the number of rows on the screen minus 1.
2) A window can not be wider than the number of columns on the screen.
3) A window can not be shorter than one row of displayable data.
4) A window can not be narrower than the width of the Window Command area (7 columns).
5) Once a boundary bar is beyond the edge of the screen, issuing further commands to move the boundary bar outwards will not take effect.

DESIGNATING A DOMINANT WINDOW

You may designate a window that is currently recessive (overlapped by another window) to be Dominant by entering the SWITCH command in the current Dominant Window.

The command is entered in the Window Command Area of the Dominant window as **WSx** where x is the window number of the recessive window to be made dominant.

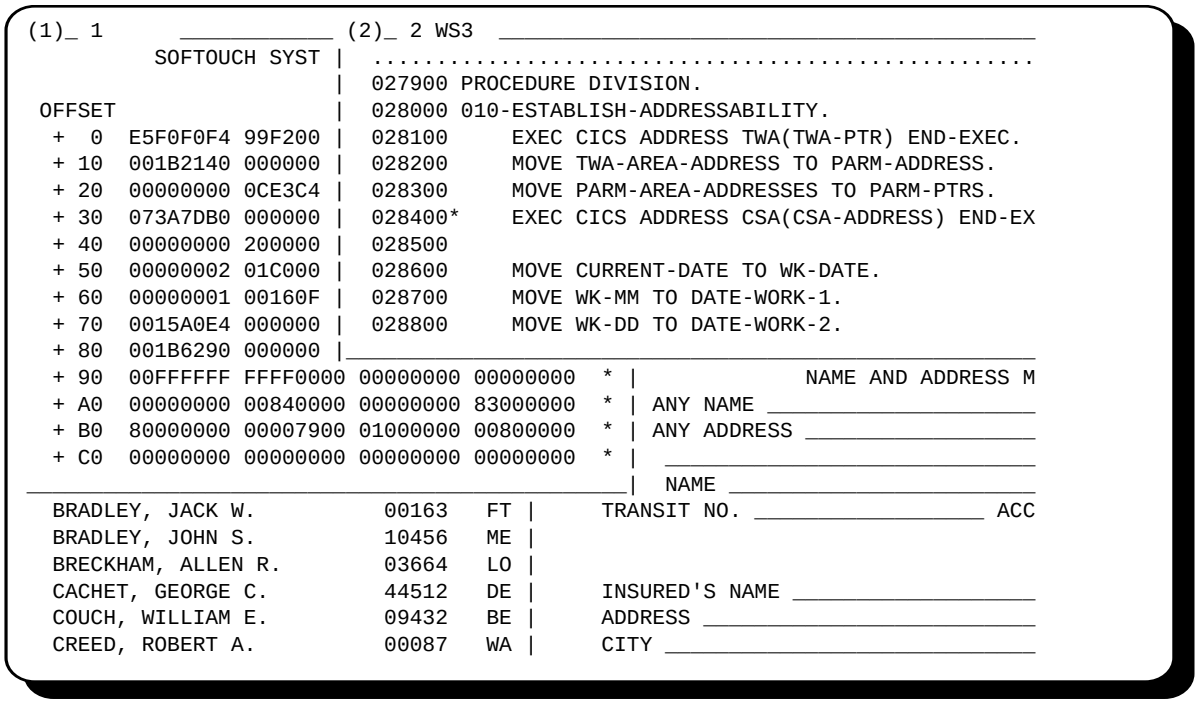
It is important to note that the SWITCH command designates a window number, not a virtual terminal number. The window number is displayed in parentheses in the left-most byte of the upper window boundary bar of each window (if you cannot see it it's because it is overlapped by another window). The number in the window command area is the virtual terminal number. For standard and variable window modes, the sequence of window numbers is always the same:

- Four-windows. Upper-left is window 1, upper-right is window 2, lower-left is window 3 and lower right is window 4.
- Three-windows. Upper-left is window 1, upper-right is window 2 and lower is window 3.
- Two horizontal windows. Upper is window 1 and lower is window 2.
- Two vertical windows. Left is window 1 and right is window 2.

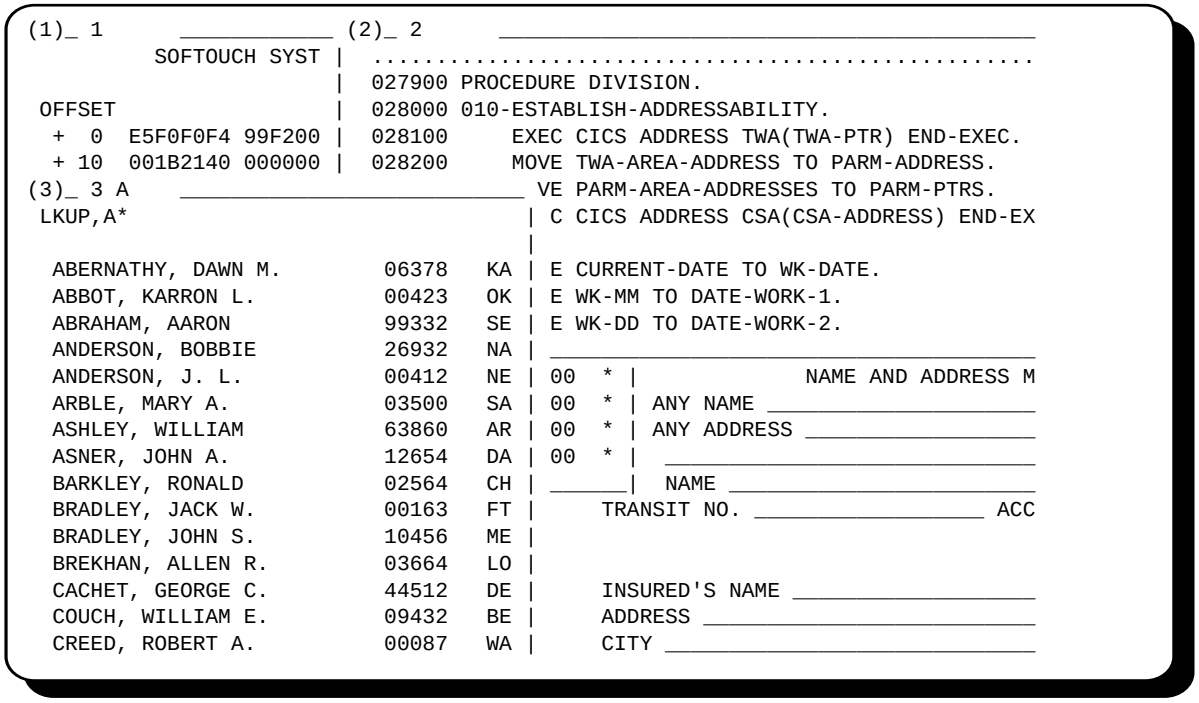
When the SWITCH Command is issued, the resultant Dominant Window becomes the active Window for transaction initiation and the cursor is positioned to that window. If the window mode is Standard, the SWITCH command will change the window mode to Variable.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In the following example, Window 2 is dominant:



If the WS3 Command were issued, Window 3 would become dominant and the display would change to:



© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WINDOW SWITCH KEY OPERATION

If Window Switch Keys have been established for this terminal, you may accomplish a WS command by pressing the PF or PA key that corresponds to the window number that is to be made Dominant.

To illustrate, using the above two examples, assume PF13 is assigned to window 1, PF14 to window 2, PF15 to window 3 and PF16 to window 4. Pressing PF15 would make window 3 the Dominant window, exactly as if a WS3 command were issued from the previous Dominant window.

CHANGING WINDOW CONFIGURATION

If you are using Standard or Variable window modes, you can change from VARIABLE to STANDARD window mode, or from STANDARD to VARIABLE window mode at any time, as well as change the number of windows.

To change from Standard to Variable window mode, enter any of the WINDOW ADJUSTMENT COMMANDS discussed above, or enter a WINDOW SWITCH command. These commands automatically place you in VARIABLE window mode.

To change from Variable to Standard window mode, you must enter a CICS-WINDOWS command, either from within a window or in full-screen mode. The command is WNDO,WIN=x

where x is the window configuration code desired. The available configuration codes are:

- 2 Two horizontal windows
- 2H Two horizontal windows
- 2V Two vertical windows
- 3 Three windows
- 4 Four windows

You must have at least as many virtual terminals available as the number of windows requested.

This command can be entered from any CICS-WINDOWS virtual terminal, from a clear screen or window.

Upon entry of the WNDO,WIN=x command, the requested number of windows is allocated for your terminal and the terminal is placed in STANDARD window mode. Note that if all you want to do is change from variable to standard window mode and keep the same window configuration, simply enter the WNDO,WIN=x command requesting the same number of windows as is currently in use.

STANDARD, VARIABLE AND POPUP WINDOW MODES

Operationally, it is important to understand the difference in STANDARD, VARIABLE and POPUP windows. They operate somewhat differently, and each method provides certain advantages, depending on your application needs. The following discussion deals first with STANDARD windows, then VARIABLE windows, then finally POPUP windows.

STANDARD WINDOW CONFIGURATIONS

The standard window configuration is dependent upon the number of windows and the vertical or horizontal option selected by the user or as pre-defined in the customization table (see section 11 - CUSTOMIZATION).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

There are four variations of Standard Windows. These are:

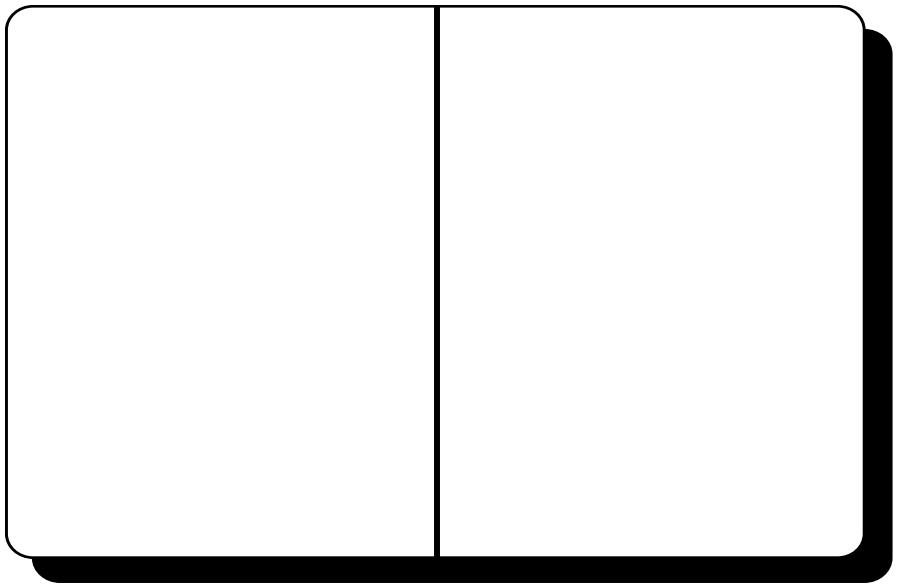
1) Two horizontal windows. This is a normal "split" screen, divided into two sections, each window extending the full width of the screen.
2) Two vertical windows. In this mode, the screen is split vertically, each window extending the full depth of the screen, less one row for the window command line.
3) Three windows. This mode has two small quadrant windows at the top and a half-screen horizontal window at the bottom.
4) Four windows. This configuration consists of four quadrant windows.

Following are examples of the four Standard Window Configurations. (For a discussion of methods of varying the window configurations, see VARIABLE WINDOW CONFIGURATIONS later in this section).

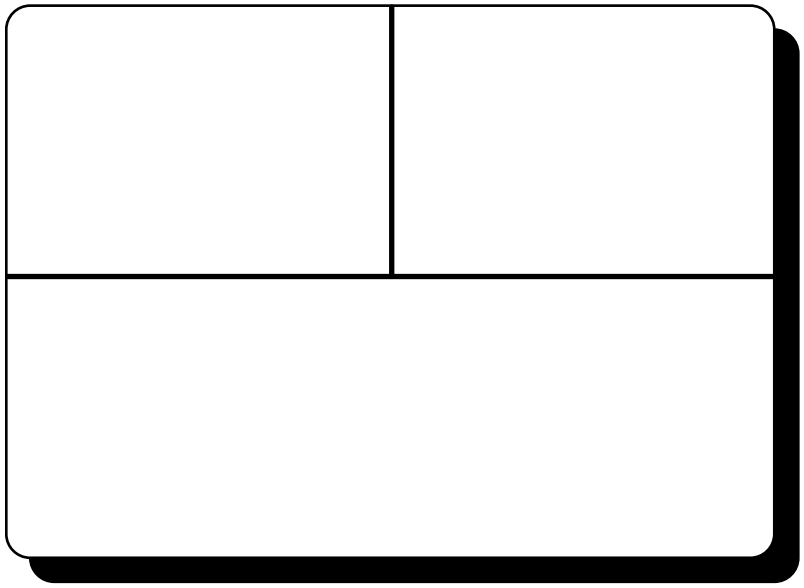
A two-horizontal window display will look like this:

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

A two-vertical window display will look like:

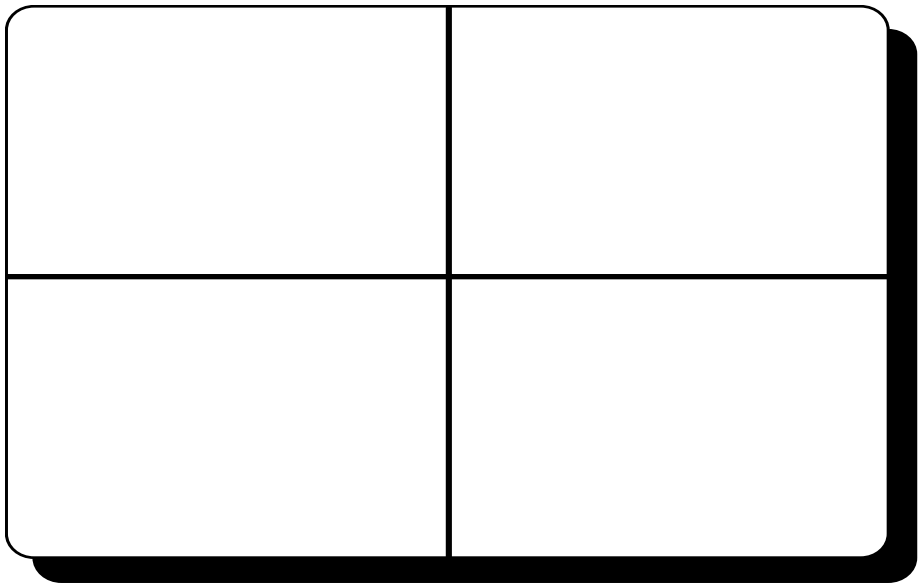


A three-window display will look like:



© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

A four-window display will appear as follows:



The default size of each standard window depends upon the type of terminal you are using. Different terminal models allow different maximum screen sizes and CICS-WINDOWS will take advantage of that fact to give you the largest possible windows. The following table describes the window sizes for each of the four current terminal types that are available:

Terminal Model No.	Maximum Screen Size		Horizontal Window		Vertical Window		Small Window	
	Rows	Columns	Rows	Columns	Rows	Columns	Rows	Columns
2	24	80	11	80	23	38	11	38
3	32	80	15	80	32	38	15	38
4	43	80	21	80	43	38	21	38
5	27	132	11	132	26	64	12	64

OPERATION WITH STANDARD WINDOWS

In STANDARD window mode, data may be entered in any or all windows while in window mode. It is possible to update fields in more than one window at a time. To actually process the application, that is, send the complete transaction to the application system, the user must "activate" the window by placing an "A" in the window command area. (For more information, please refer to RESPONDING TO A WINDOW later in this section).

To illustrate: A file maintenance transaction is displaying a record on the screen and waiting for data fields to be keyed (a record is being updated or added to the file). The operator may switch to window mode in order to simultaneously view data from another application. While still in window mode, the new data may be keyed into the data fields. Scrolling may be performed if necessary. When all the data has been entered, the operator activates the window by placing an "A" in the window command area and pressing ENTER. The data is updated according to the specifications of the application.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Or, as an alternative, the operator may input the data while in window mode and then switch to full-screen mode. All of the new data which has been keyed will remain in the fields upon return to full-screen mode. The application may be processed by pressing ENTER or whatever PF key is required.

RESPONDING TO A WINDOW - STANDARD WINDOWS

When you wish to initiate an application from one of the windows in STANDARD window mode, you must:

1) Tab to the command input area above the desired window.
2) Enter an "A" (active window) as the command in the window control line.
3) Enter the transaction data or command according to the normal procedure for the application. Note that the first row following the window control line area is recognized as row one for the application. Thus if a transaction code which would normally be entered in row one, column one of a full screen is to be entered, it is entered on the first row below the window control line.
4) If the application field requiring input is not displayed in the window, you simply scroll the window until the required field is in view. It is not necessary, however, that the window be panned to top-left (equivalent to row one, column one) when ENTER (or PF key) is pressed
5) When all transaction input has been completed, be sure that the "A" is in the command area of the window and press ENTER (or a PF key, as appropriate) to process the data. If you failed to place the "A" in the window control line, nothing is lost. However, you must key in the "A" command and then press ENTER or the PF key again in order for the application to process the data Note that if the response to the application in the window is a PA key, just pressing the PA key will not work. You must enter 'PA1', 'PA2', or 'PA3' in the windows command area and press ENTER
6) The updated display for that application will return in the same window, with all other windows unchanged. The "A" remains in the window control line of the active window, and subsequent responses to the same window may be entered without referencing the window control line again

CLEARING A WINDOW - STANDARD WINDOW

In STANDARD window mode, to clear a single window while leaving the remaining windows intact, enter a "C" in the command area of the window to be cleared and press ENTER. A new transaction may then be initiated in that window by entering the transaction code. The 'C' command emulates the CLEAR key in full-screen mode. If an application normally gets re-invoked when CLEAR is pressed, the 'C' command will re-invoke the application passing it the CLEAR key just as if it were running in full-screen mode.

Note that pressing the CLEAR key in STANDARD window mode will not erase the entire screen. The windowed screen is simply re-displayed.

DIRECT SESSION KEY OPERATION IN STANDARD WINDOW MODE

If Direct Session Keys have been established for this terminal in the customization table (see Direct Sessions Key in section 11 - CUSTOMIZATION) they function as follows when in STANDARD Window Mode:

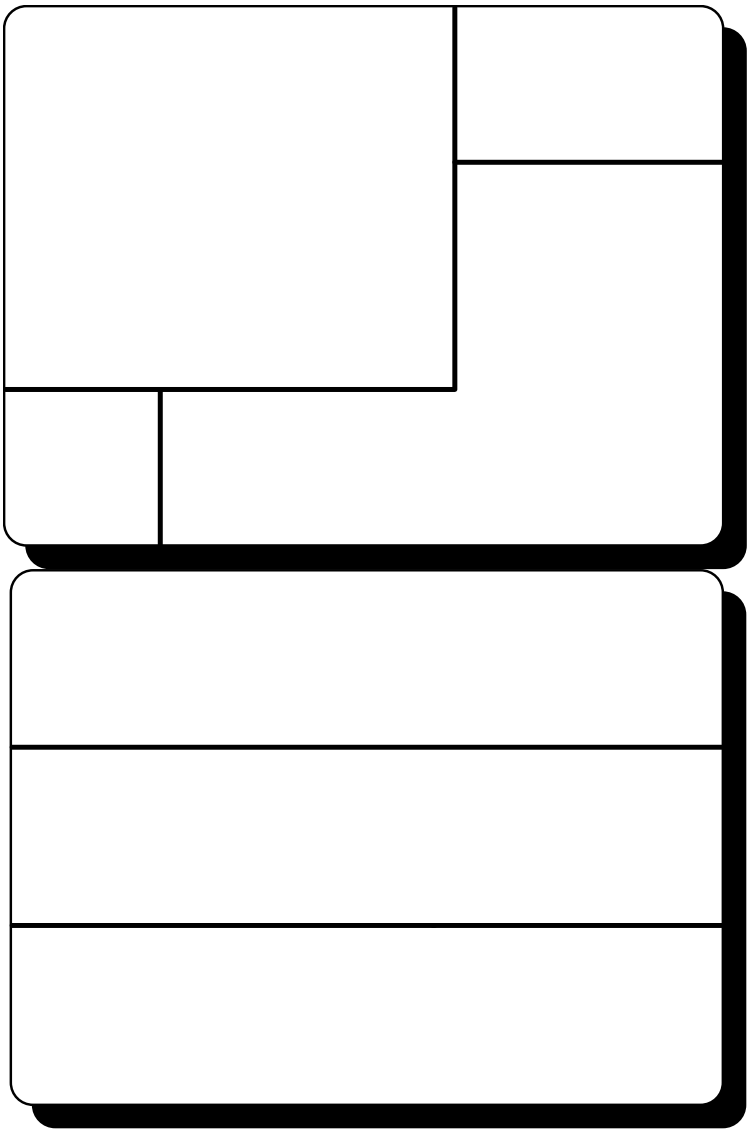
Pressing a Direct Session Key will retrieve the application for the requested virtual terminal and display it in the upper left-hand window (window number 1). All other windows will display session data in sequence relative to window 1.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

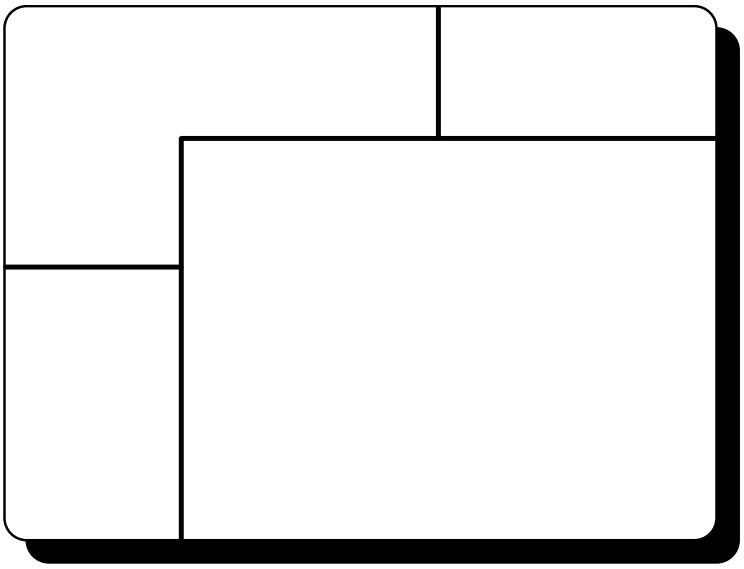
VARIABLE WINDOW CONFIGURATIONS

Once in window mode, any or all of the standard windows can be made larger or smaller, wider or taller. This is accomplished through a series of window mode commands known as WINDOW ADJUSTMENT COMMANDS.

There are hundreds of possible window configurations that can be set using variable windows. A few examples, using three or four windows might appear as follows:



© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0



OPERATION WITH VARIABLE WINDOWS

Variable Window Configuration allows overlapping windows where one window is considered Dominant and all other windows are Recessive.

In this mode, only the Dominant window contains any unprotected fields. Recessive windows are all protected, which means that the tab key will not skip the cursor into any of those windows. Thus, the only window that can be modified is the Dominant Window.

In order to modify a different window you must issue The Window-Switch Command to make a different window Dominant. See the discussion entitled DESIGNATING A DOMINANT WINDOW earlier in this section, for more information on the window-switch command.

RESPONDING TO A WINDOW - VARIABLE WINDOWS

When you wish to initiate or respond to an application from a window in VARIABLE window mode, you must:

Enter the transaction data or command according to the normal procedure for the application. Note that the first row following the window control line area is recognized as row one for the application. Thus if a transaction code which would normally be entered in row one, column one of a full screen is to be entered, it is entered on the first row below the window control line.

If the application field requiring input is not displayed in the window, you simply scroll the window until the required field is in view or use one of the WINDOW ADJUSTMENT COMMANDS (described earlier) to enlarge the window. It is not necessary that the window be panned to top-left (equivalent to row one, column one) when ENTER (or a PF key) is pressed.

When all transaction input has been completed, press ENTER (or a PF key, as appropriate) to process the data. The "A" in the window control line is always present when in VARIABLE window mode.

Note that if the response to the application in the window is a PA key, unlike Standard window mode, you do not need to enter 'PA1', 'PA2', or 'PA3' in the window command area, just pressing the PA key will work.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The updated display for that application will return in the same window, with all other windows unchanged. The "A" remains in the window control line of the dominant window, and subsequent responses to the same window may be entered without referencing the window control line again.

OVERLAPPING WINDOWS

It is important to understand one fundamental difference between Standard Window configurations and Variable Window configurations. Once you alter the size of a window, that window could partially cover, or overlap at least one and sometimes more than one other window.

When this happens, the overlapped window still remains the same size that it was before the overlap occurred. You simply cannot see the portion of data that is overlapped.

When a Window Switch command to the overlapped window is executed, the window will rise to the top of the previously overlapping window(s).

CLEARING A WINDOW - VARIABLE WINDOWS

When using VARIABLE window mode, pressing the CLEAR key will erase the dominant window, and if needed, re-invoke the application program in that window, passing it the CLEAR key.

You may also use the 'C' command in the window command area to clear it if desired.

DIRECT SESSION KEY OPERATION IN VARIABLE WINDOW MODE

If you are using Variable Window configurations, pressing a Direct Session Key will retrieve the application for the requested virtual terminal and display it in the Dominant window. All other windows will display session data in sequence relative to the Dominant window.

METHODS OF OPERATION USING VARIABLE WINDOWS

There are several methods of operation in Variable Window mode:

- You may designate a particular window as the Dominant Window by using the window command known as the SWITCH Command or the Window Switch keys (see WINDOW ADJUSTMENT COMMANDS, later in this section).
- You may establish a Dominant Window and rotate the virtual terminals through that window by pressing the toggle-forward or toggle-backward key (see TOGGLING WHILE IN WINDOW MODE, earlier in this section).
- You may establish a Dominant Window and move the virtual terminals directly to that window by pressing the Direct Session Key for the desired virtual terminal (see TOGGLING WHILE IN WINDOW MODE, earlier in this section).

You may now operate a transaction exactly as if it were in full-screen mode. The cursor will not tab out of the dominant window. If you are using a color terminal, only the Dominant window will display different colors, the Recessive windows will display all fields in blue.

OPERATION WITH POPUP WINDOWS

Popup windows operate much the same as variable windows with the following exceptions:

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In variable window mode, any given window has at least two edges tied to a corner of the screen. In popup mode, all windows are free floating. In other words, any window may appear anywhere on the screen. This difference is better portrayed in *WINDOW EXAMPLES*, later in this section.

Also in variable window mode, regardless of the number of sessions, the user may have only four windows. However with popup windows, the user may have up to nine windows, and the user may specify how each window is to be displayed while the window is inactive. This is explained in more detail below, in *ACTION BAR FUNCTIONS OF POPUP WINDOWS* under ICON, HIDE, and FULL functions.

In addition to the Window Switch command used in Variable window mode, to perform the same function in Popup window mode, you may move the cursor to a window and press ENTER. The window to which the cursor was pointing when ENTER was pressed will then become the dominant window.

Furthermore, POPUP windows have an action bar, which enables an operator to perform windowing functions without the need to memorize windowing commands or to tie up PF keys for lesser used functions.

Popup windows also take advantage of extended color and highlighting when displaying window borders (the lines that surround each window) to more distinctly separate windows. For this feature to function, the terminal must support extended color and/or highlighting and the EXTDS, COLOR and/or HILIGHT features must be present in the DFHTCT macro definition for the terminal. In addition, graphic escape characters may be used to display the window border.

ACTION BAR FUNCTIONS OF POPUP WINDOWS

POPUP windows differ from Variable and Standard window modes, in that they allow the use of an action bar to perform functions such as:

- Altering the size and position of the window
- Altering the display attributes while the window is recessive.
- Exiting window mode
- Switching to a different window
- Toggling to a different session

The following discussion explains the available options of the action bar:

WINDOW

MOVE

This is used to move the window. To use this function simply tab to the function and press ENTER. The function should now be highlighted. Now position the cursor to an area on the screen where you wish to place the top-left corner of the window, then press ENTER. The screen will redisplay with the upper-left corner of the window at the position of the cursor when ENTER was pressed.

SIZE

This is used to alter the size of the window. To use this function, tab to the function and press ENTER. The function should now be highlighted. Now position the cursor to the area on the screen where you wish to place the bottom-right corner of the window, then press ENTER. The screen will redisplay with the lower-right corner of the window at the position of the cursor when ENTER was pressed.

ICON [ON | OFF]

This is used to specify that this window is to be displayed as an icon at the bottom of the screen while this window is recessive. If 'ICON ON' is displayed, this means that if you tab to the option and press ENTER, the Icon setting will be turned on. In other words, 'ICON ON' is not specifying that the Icon setting is on, it

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

is specifying that if this option is selected, CICS-WINDOWS will perform the function of turning on the Icon setting.

[Note] Icon, Hide and Full are mutually exclusive items. That is, if one is turned on the other two will be turned off. However, all three may be turned off. In this case, the window will display as it appears while dominant, except as low-intensity and protected.

HIDE [ON | OFF]

This is used to specify that this window is not to be displayed on the screen while this window is recessive. This function operates in the same manner as ICON, above.

FULL [ON | OFF]

This is used to specify that this window is to be displayed as a full screen in the background while this window is recessive. This function operates in the same manner as ICON, above.

EXIT

EXIT WINDOW MODE

This function is the equivalent of pressing the Window key, to exit.

SWITCH TO WINDOW

This may be used as an alternative to window switch keys. When this option is selected, a popup window will appear showing information about each window, such as session number, virtual terminal number and active transactions. To perform the window switch, tab the cursor to the desired window number and press ENTER.

TOGGLE TO SESSION

This may be used as an alternative to the toggle keys or sessions direct keys. When this option is selected, a popup window will appear showing information about each session such as window number, virtual terminal number and active transactions. To perform the toggle, tab the cursor to the desired session number and press ENTER.

DISPLAY CONTROL WINDOW

This option is used to display the Control Window. For more information on the Control Window, see section 02 - OPERATION under THE CICS-WINDOWS CONTROL WINDOW.

WINDOW EXAMPLES

To illustrate Variable and Popup window modes with Dominant and Recessive Windows, assume we have four virtual terminals established, using four windows, with the following transactions active in each virtual terminal:

Terminal 1 - A hexadecimal memory display.

Terminal 2 - An editor with a program displayed.

Terminal 3 - An alphabetic name search.

Terminal 4 - A File maintenance transaction.

In full-screen mode, the four transaction displays appear as follows:

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Virtual Terminal 1

```
         SOFTOUCH SYSTEMS INC. 1.0 01/01/91 MEMORY DISPLAY/ALTER

 OFFSET                                                              ADDRESS
  +  0  E5F0F0F4 99F20006 001B0E50 001B0E50  * V004r2.....&...& *  00159F48
  + 10  001B2140 00000000 00166607 30000000  * ................ *  00159F58
  + 20  00000000 0CE3C440 FFFFFF78 00000990  * .....TD......... *  00159F68
  + 30  073A7DB0 00000000 00000000 00000000  * ..'............. *  00159F78
  + 40  00000000 20000000 07801850 00000000  * ...........&.... *  00159F88
  + 50  00000002 01C00000 00000000 00000000  * ................ *  00159F98
  + 60  00000001 00160F9C 001990E4 00000000  * ...........U.... *  00159FA8
  + 70  0015A0E4 00000000 00000000 00000000  * ...U............ *  00159FB8
  + 80  001B6290 00000000 0015F1B8 00000000  * ...0......1..... *  00159FC8
  + 90  00FFFFFF FFFF0000 00000000 00000000  * ................ *  00159FD8
  + A0  00000000 00840000 00000000 83000000  * .....d......c... *  00159FE8
  + B0  80000000 00007900 01000000 00800000  * e.....=......... *  00159FF8
  + C0  00000000 00000000 00000000 00000000  * ................ *  0015A008
  + D0  0B000000 100307D1 00000000 00000000  * .......J........ *  0015A018
  + E0  00000000 00000000 00000000 00000000  * ................ *  0015A028
  + F0  3A008400 007E0080 00000000 00000000  * ..d..c.e........ *  0015A038


PF1=HELP         PF3=EXIT         PF5=PREV COMMAND
PF7=BACKWARD     PF8=FORWARD      PF9=UPPERCASE ON          PF10=STS PRODUCTS
 ==> TCT=*
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Virtual Terminal 2

```
.....................................................................
027900 PROCEDURE DIVISION.
028000 010-ESTABLISH-ADDRESSABILITY.
028100     EXEC CICS ADDRESS TWA(TWA-PTR) END-EXEC.
028200     MOVE TWA-AREA-ADDRESS TO PARM-ADDRESS.
028300     MOVE PARM-AREA-ADDRESSES TO PARM-PTRS.
028400*    EXEC CICS ADDRESS CSA(CSA-ADDRESS) END-EXEC.
028500
028600     MOVE CURRENT-DATE TO WK-DATE.
028700     MOVE WK-MM TO DATE-WORK-1.
028800     MOVE WK-DD TO DATE-WORK-2.
028900     MOVE WK-YY TO DATE-WORK-3.
029000     MOVE DATE-WORK TO CUR-DATE.
029100     ACCEPT WK-TIME FROM TIME.
029200     MOVE WK-HR TO DATE-WORK-1.
029300     MOVE WK-MIN TO DATE-WORK-2.
029400     MOVE WK-SEC TO DATE-WORK-3.
029500     MOVE DATE-WORK TO CUR-TIME.
029600
029700     IF HUX-WNDO-MODE = 'L'
029800         GO TO 1000-LETTER.
029900     GO TO 9000-NOT-LIST-MODE.
030000     EJECT
030100*************************************************************
```

Virtual Terminal 3

```
LKUP,A*

 ABERNATHY, DAWN M.        06378   KANSAS CITY, MO
 ABBOT, KARRON L.          00423   OKLAHOMA CITY, OK
 ABRAHAM, AARON            99332   SEATTLE, WA
 ANDERSON, BOBBIE          26932   NASHVILLE, TN
 ANDERSON, J. L.           00412   NEW YORK, NY
 ARBLE, MARY A.            03500   SAN DIEGO, CA
 ASHLEY, WILLIAM           63860   ARDMORE, OK
 ASNER, JOHN A.            12654   DALLAS, TX
 BARKLEY, RONALD           02564   CHICAGO, IL
 BRADLEY, JACK W.          00163   FT. WORTH, TX
 BRADLEY, JOHN S.          10456   MEMPHIS, TN
 BREKHAN, ALLEN R.         03664   LOS ANGELOS, CA
 CACHET, GEORGE C.         44512   DETROIT, MI
 COUCH, WILLIAM E.         09432   BETHANY, OK
 CREED, ROBERT A.          00087   WASHINGTON, DC
 DANELY, JAMES L.          74811   OKLAHOMA CITY, OK
 DOVER, EARL III           36029   LUNAR SURFACE
 DUGAN, MAX L.             98473   NEW YORK, NY
 DUNLAP, SAMANTHA          34801   MELBOURNE, AUST
 DUNN, RALPH K.            09358   SAN DIEGO, CA

PF3=EXIT   PF7=BACKWARD   PF8=FORWARD
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Virtual Terminal 4

```
                    NAME AND ADDRESS MAINTENANCE
    COMPANY NAME ____________________________________________________________
    COMPANY ADDRESS _________________________________________________________
    CITY ____________________________ STATE ___________ ZIP _________
    BANK NAME _______________________________________________________________
    TRANSIT NO. ____________________ ACCOUNT NUMBER ____________________


    INSURED'S NAME __________________________________________________________
    ADDRESS _________________________________________________________________
    CITY ____________________________ STATE ___________ ZIP _________


          KEY ALL REQUIRED FIELDS AND PRESS ENTER TO ADD RECORD
```

In a Standard 4-windows configuration, the display would appear as follows:

```
(1)_ 1 A    _______________________________ (2)_ 2      _____________________________
          SOFTOUCH SYSTEMS INC. 1.0 01 |  .....................................
                                       |  027900 PROCEDURE DIVISION.
 OFFSET                                |  028000 010-ESTABLISH-ADDRESSABILITY.
  +  0  E5F0F0F4 99F20006 001B0E50 001 |  028100     EXEC CICS ADDRESS TWA(TWA-
  + 10  001B2140 00000000 00166607 300 |  028200     MOVE TWA-AREA-ADDRESS TO P
  + 20  00000000 0CE3C440 FFFFFF78 000 |  028300     MOVE PARM-AREA-ADDRESSES T
  + 30  073A7DB0 00000000 00000000 000 |  028400*    EXEC CICS ADDRESS CSA(CSA-
  + 40  00000000 20000000 07801850 000 |  028500
  + 50  00000002 01C00000 00000000 000 |  028600     MOVE CURRENT-DATE TO WK-DA
  + 60  00000001 00160F9C 001990E4 000 |  028700     MOVE WK-MM TO DATE-WORK-1.
  + 70  0015A0E4 00000000 00000000 000 |  028800     MOVE WK-DD TO DATE-WORK-2.
(3)_ 3      _______________________________ (4)_ 4      _____________________________
 LKUP,A*                               |                     NAME AND ADDRESS M
                                       |     COMPANY NAME _____________________
  ABERNATHY, DAWN M.        06378   KA |     COMPANY ADDRESS __________________
  ABBOT, KARRON L.          00423   OK |     CITY _____________________________
  ABRAHAM, AARON            99332   SE |     BANK NAME ________________________
  ANDERSON, BOBBIE          26932   NA |     TRANSIT NO. ____________________ ACC
  ANDERSON, J. L.           00412   NE |
  ARBLE, MARY A.            03500   SA |
  ASHLEY, WILLIAM           63860   AR |     INSURED'S NAME ___________________
  ASNER, JOHN A.            12654   DA |     ADDRESS __________________________
  BARKLEY, RONALD           02564   CH |     CITY _____________________________
```

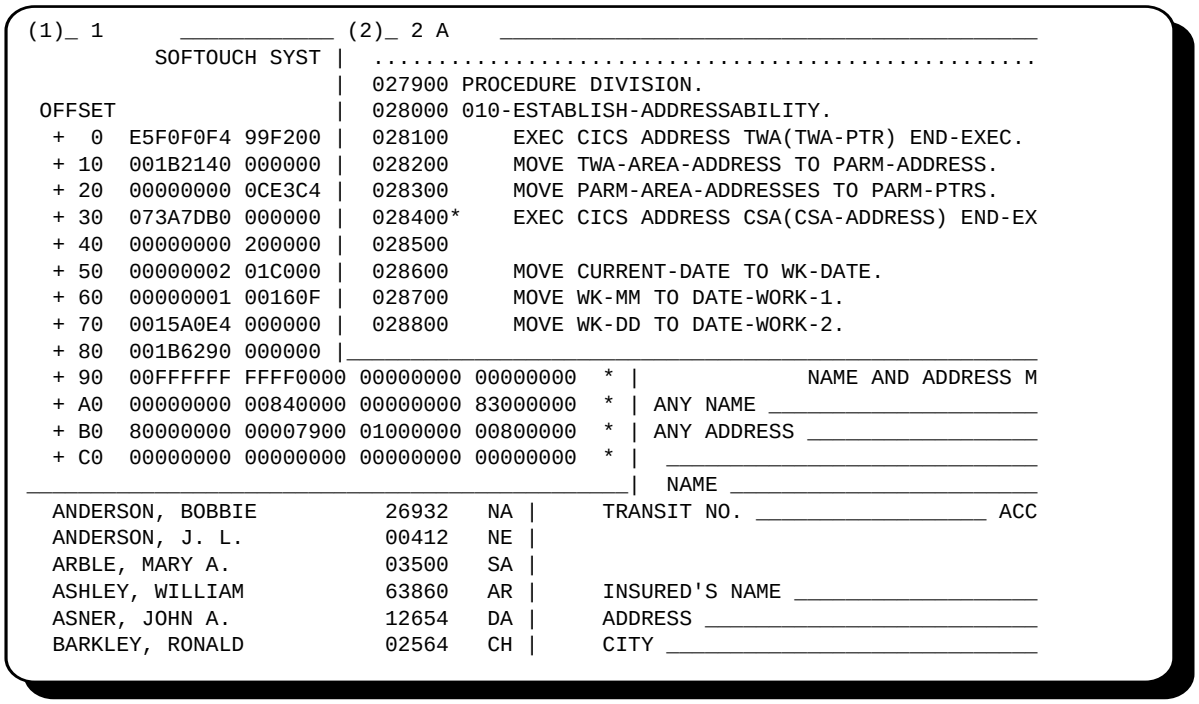
Each window has its own command line with its number in parentheses. Each virtual terminal is represented by the number without parentheses.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Since this is a standard window display, all of the windows carry the attributes of the portion of the application screen that is displayed in the window. If you press the TAB key, the cursor will move from window to window as it finds unprotected fields.

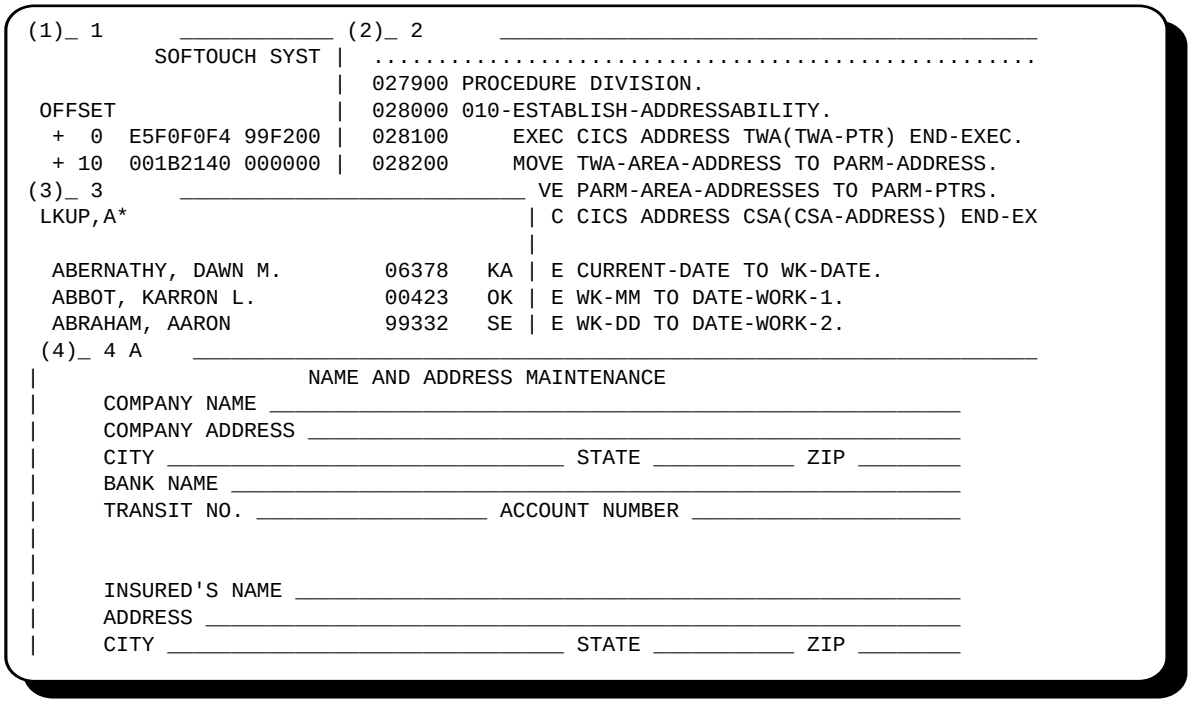
If this were a variable window display, as in the following examples, only the dominant window would carry the true attributes of the application. All the other windows would be completely protected and the cursor would not tab out of the dominant window.

In the following example the windows have been adjusted in size and shape. Window (2) is the Dominant Window. Window (2) overlaps (1) and (1) overlaps both (3) and (4).

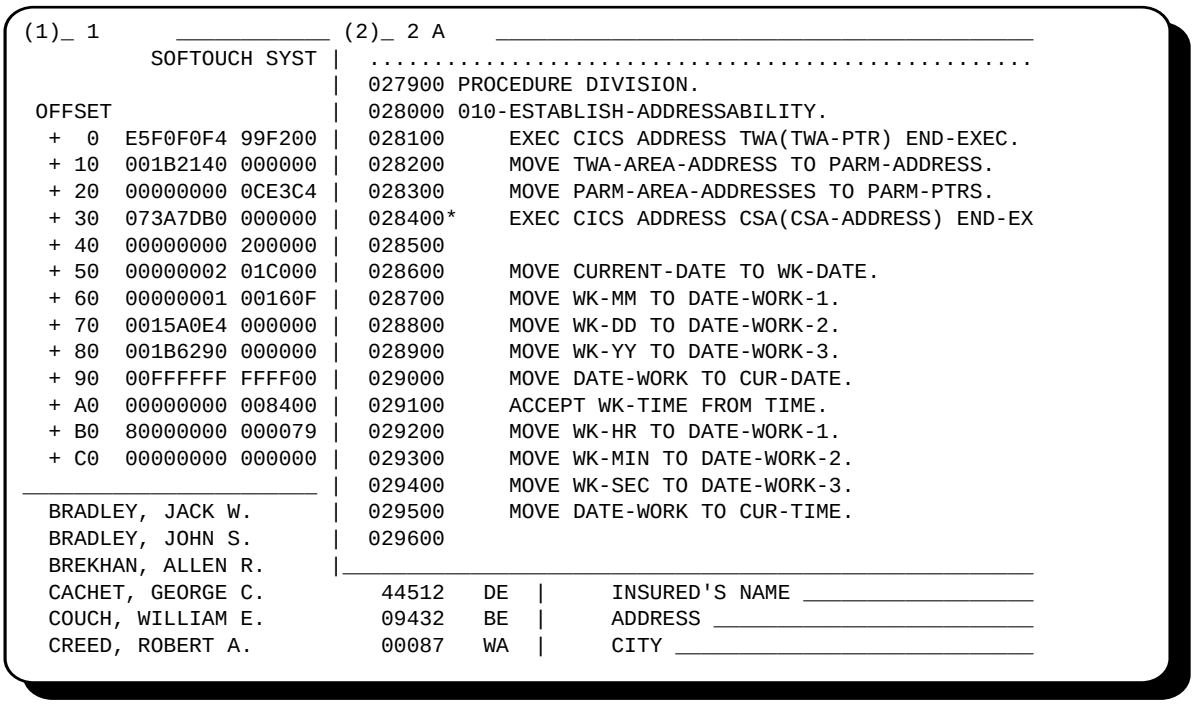


© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In this example, Window (4) is the Dominant Window. Window (3) overlaps (1) and (2), and (4) overlaps (3).

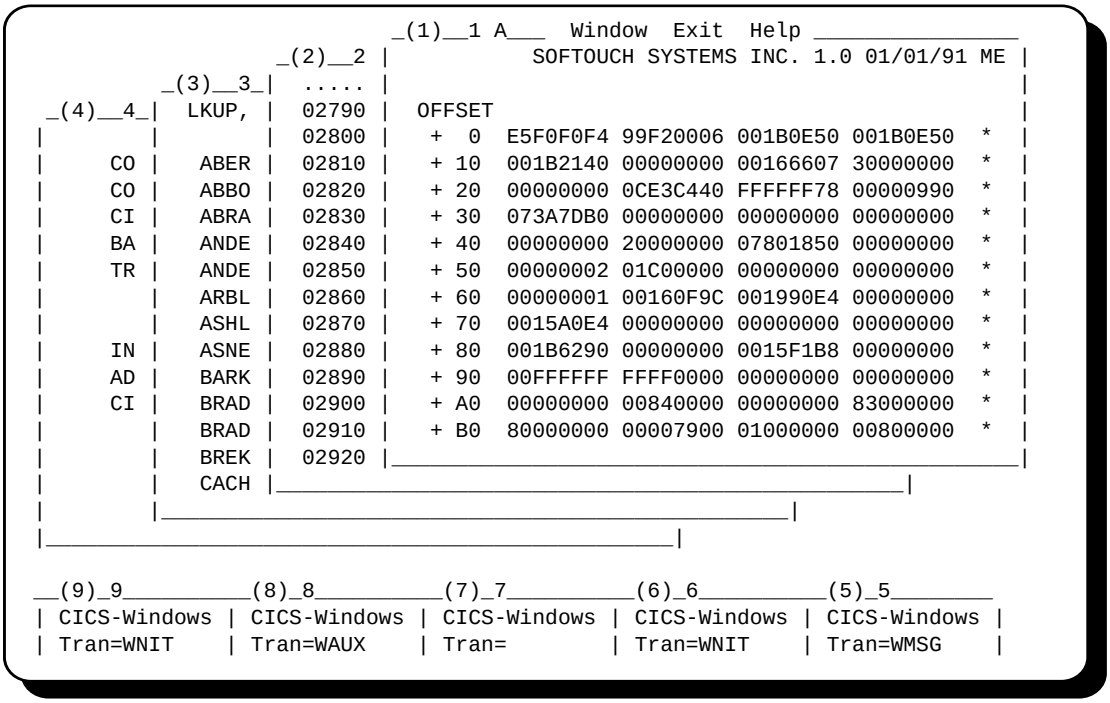


In the next example, Window (2) is again Dominant but it has been made larger by moving the lower bar down. Now Window (2) overlaps all three of the other windows.

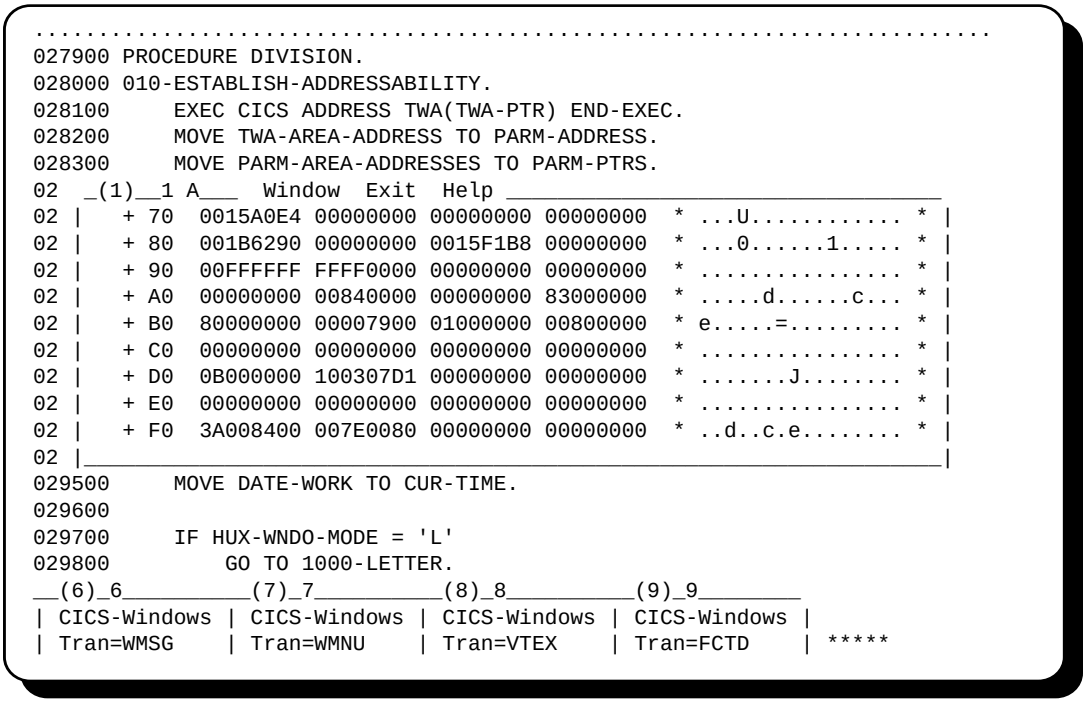


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

This concludes the examples for Variable windows. Following are examples for Popup windows. This example shows 9 popup windows in the default configuration.

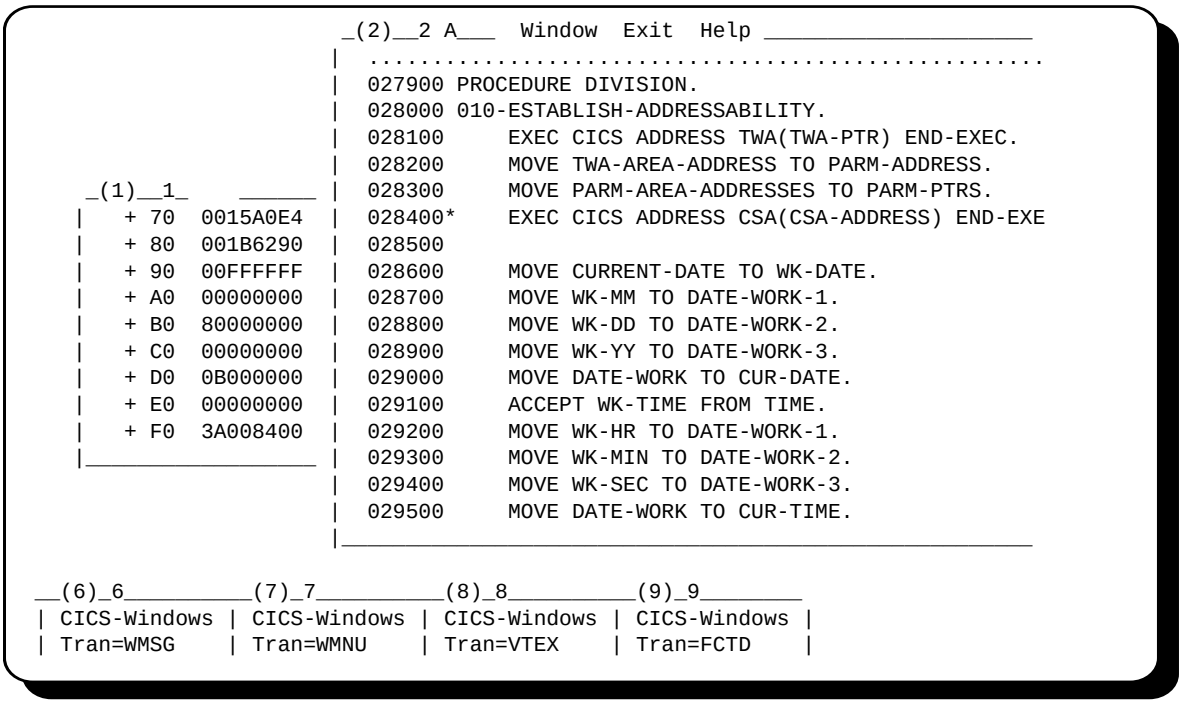


The following example shows the display attributes for windows while they are recessive. In this example, window one is the dominant window, window 2 has the Full attribute, windows 3, 4, and 5 have the Hide attribute, and windows 6 through 9 have the Icon attribute. (Notice the windows with the Icon attribute have the current transaction code displayed in the window.)



© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In this example, a Window Switch to window 2 has been performed. In the previous example, window 2 had the Full screen attribute. Now that window 2 is active, it shape has returned to the same size and position when the window was last dominant.



Note that window 1 remained as it appeared while active. This is because no recessive display attribute was selected for this window. If no attribute is selected, the window will remain in place maintaining its size, shape, and position. In other words, the window will appear as it would while using Standard or Variable window mode.

As you can see from these examples, the possibilities for different window combinations are many. Through the use of Variable or Popup Window Configurations you can set up your terminal with a "best-fit" configuration for the screen displays of the transactions that you most frequently use.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 4. EXAMPLES OF USE

Example 1: Program development and testing.

One excellent use of CICS-WINDOWS is in the area of on-line (or batch) program development. The advantage is seen in setting up three separate windows during application development and testing. Window One could contain the application program text editor. Window Two could display the data base. Window Three could be where the application is run. CICS-WINDOWS eliminates the need for logging in and out of the text editor as changes are made. It also eliminates the need to fetch the application and to position to the appropriate location. During application development you watch Window Three to see the application run while Window Two shows the results. As changes are required you toggle over to Window One to use the text editor.

Example 2: Having the Menu Available on the Same Screen as the Transaction

Some CICS applications are menu-driven so that you must first make selections from one or more menu screens before you may enter an application. This makes it quite tedious to transfer from one transaction to another. This problem is easily solved with CICS-WINDOWS because you are able to view the menu in one window while viewing the transaction in the other window.

Example 3: Using Data from Multiple Applications.

With CICS-WINDOWS you can be far more productive by having the inquiry transaction in Window One and the input transaction in Window Two. It then becomes a simple matter of viewing the information in Window One and entering that information into the transaction in Window Two.

To illustrate: Suppose you have an application which requires input data that is available in an entirely different application transaction. Without CICS-WINDOWS you would have to bring up the transaction which contains the data you seek and then write down the information or print it out. After doing all that extra work you would still have to return to your original transaction to enter in the new data and then update. With CICS-WINDOWS you only have to shift your eyes to a different window to obtain the data you need. Furthermore, you may use the cut and paste feature of CICS-WINDOWS to copy the fields. This all but eliminates any operator keying errors, and frustration when keying account numbers or other unintelligible strings of characters.

Example 4: Data Entry Exiting.

Many data-entry applications and packages keep running totals of records entered, amounts, and other data. In order to exit in the middle of such a batch, most of these applications require special commands, such as SAVE, RESTORE, or RESTART-BATCH. With CICS-WINDOWS the data-entry operator can move to another application, while either in a full screen or window mode, and then return to the entry screen without disturbing the data-entry application in any way.

Example 5: Saving your place in a multi-system environment.

When CICS is only one of several on-line systems on the machine, or when multiple CICS systems are in use, it may be desirable to save the current application status prior to exiting CICS altogether. This can be accomplished with CICS-WINDOWS.

To illustrate: Suppose a programmer in an OS system is testing a CICS transaction, and needs to log off of CICS in order to go to TSO to edit the program. If the programmer "toggles" out of the current CICS transaction before signing off of CICS, the application in progress will be saved. After signing back on to CICS, a single keystroke of the "toggle" key will refresh the screen to exactly as it was when the screen was left.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Note that this is one of the features of CICS-WINDOWS which can be deactivated by the User Option table (see the discussions on *REQUIRE WNDO,OFF AT LOGOFF, REQUIRE CLEAR SESSIONS BEFORE OFF, REQUIRE TRANSACTION END BEFORE OFF* and *FORCE PURGE AT SIGNON AND SIGNOFF* in section 11 - *CUSTOMIZATION*).

Example 6: Toggling and Windowing Applications on Multiple CICS Systems using MRO or ISC

In an environment where more than one CICS system is run (in different regions or on different machines) CICS-WINDOWS can be used to toggle between applications running in each CICS system or simultaneously display data in a window from each CICS system.

In order to do this your CICS system must be communicating by means of the Multiple Region Operation (MRO) or the Inter-System Communication (ISC) feature of CICS. MRO is used for multiple regions of the same machine and ISC for CICS systems running on different virtual or physical machines.

CICS-WINDOWS fully supports all options of MRO and ISC, using either remote transaction definitions or the CRTE transaction.

For a discussion of the technical aspects of running CICS-WINDOWS in an MRO or ISC environment see RUNNING CICS-WINDOWS WITH MRO/ISC in section 13 - UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS. Also see CICS REQUIREMENTS in section 10 - INSTALLATION.

Summary

There are many other uses for CICS-WINDOWS in the CICS environment. In summary it can be said that if you need to invoke a different application than the one you are in and would like to not "lose your place", or if you need to retrieve and display information from several applications at once, that's what CICS-WINDOWS is all about.

The more you use CICS-WINDOWS, the more uses you will find for it and each use will bring a corresponding increase in productivity.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 5. OPERATION

INITIATING CICS-WINDOWS

The **CICS-WINDOWS** product is very easy to initiate. This would normally be performed once, at the start of your working day and before any other applications have been processed. With the Auto-Start option, the initialization can be performed automatically when you sign on to CICS. It can, of course, be initiated at any time throughout the day.

The command to initiate or start **CICS-WINDOWS** is:

WNDO,ON

When the **WNDO,ON** command is issued, CICS-WINDOWS will search for a profile to determine the operational specifications to be used on this terminal.

[Note]: If an explicit or default profile is not found, the WNDO,ON command will be treated like a WNDO,INIT command. The WNDO,INIT command is explained in detail in section 14 - *SPECIAL PURPOSE COMMANDS.*

At this point, **CICS-WINDOWS** initialization is complete and the User Configuration Display will appear.

THE CICS-WINDOWS USER CONFIGURATION DISPLAY

Upon completion of the WNDO,INIT or WNDO,ON commands or any time a WNDO,INQ or WNDO (no command) is issued, the following User Configuration Display will appear:

```
____  Save  Keys  Exit(X)  Help               CICS-WINDOWS Release 3.2.xxxxxx
--------------------------------------------------------------------------------
                    CICS-Windows User Configuration

Make changes and press "ENTER" to alter user configuration.
Profile id            ____          Window key     ____    Control key     ____
Number of sessions    _             Help key       ____
Number of windows     _             Toggle forward ____    Toggle backward ____
Window configuration  _____         Switch forward ____    Switch backward ____
Extended attributes   _________     Scroll up      ____    Scroll down     ____
Dominant color/hilite ___ __        Scroll left    ____    Scroll right    ____
Graphic escape        ___           Scrolling rows __      Scrolling cols  __
Sessions          1      2      3      4      5      6      7      8      9
  Pseudo ids     ____   ____   ____   ____   ____   ____   ____   ____   ____
  Direct keys    ____   ____   ____   ____   ____   ____   ____   ____   ____
Windows
  Switch keys    ____   ____   ____   ____   ____   ____   ____   ____   ____
  Start row/col  __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___
  Nmbr rows/cols __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___
  Disposition    ______ ______ ______ ______ ______ ______ ______ ______ ______
  Color/Hilight  ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __ ___ __


Enter F1=Help F3=Exit F4=Save F5=Keys
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Initially the fields of the display will correspond to the fields of the User Profile that is active for this terminal (or will correspond to the responses made during the WNDO,INIT process). Some values can be changed, and once alterations to this screen are performed, those changes will remain in effect until WINDOWS is deactivated on this terminal.

FIELDS OF THE USER CONFIGURATION DISPLAY

The fields of the User Configuration display are described below. For information on operating the Action Bar, see *OPERATION OF ACTION BARS*, later in this section.

CONTROL KEY This is used to specify a PF or PA key to be pressed that will invoke the CICS-WINDOWS Control Window. For more information on the Control Window, see *THE CICS-WINDOWS CONTROL WINDOW*, later in this section.

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

DOMINANT COLOR/HILITE

If EXTENDED ATTRIBUTES is coded 'YES', 'COLOR' or 'HIGHLIGHT', this specifies the type of border that is to be used around the dominant, or active window when in window mode.

Two fields are present. The first field is the color of the window border of the dominant window. Valid entries are:

(BLU)e, (GRE)en, (PIN)k, (RED), (TUR)quiose, (WHI)te, (YEL)low
The first three characters of the color that is desired.

NO The window borders of the dominant window are to display as the same color specified in the WINDOW COLOR/HILITE field for that window.

The second field describes the type of highlighting to be used for the dominant window. Valid entries are:

IN The window borders are not to display (invisible).

NO The window borders are to remain normal.

RB The window borders are to be displayed as reverse video.

EXTENDED ATTRIBUTES

If this terminal supports extended color and/or highlighting, this specifies the type of border that is to be used around each window when in window mode. Valid entries are:

NO The window borders are to display as vertical bars and underscores. Extended color and reverse video will not be used.

COLOR The windows will assume the colors specified in the WINDOW COLOR HILIGHT fields. Reverse video borders will not be used.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

HIGHLIGHT

The window borders are to display in reverse video if specified in the WINDOW COLOR/HILIGHT fields. Extended color will not be used.

YES The window borders are to display in reverse video. Extended color and highlighting will be used.

[Note]: For extended attribute borders to be used permanently, the CICS TCT definition must specify support for EXTENDEDDS, COLOR, and/or HILIGHT and this field must be coded in the User Profile. If the TCT does not have these values coded, but you believe the terminal will support extended datastreams, you can set this option while in window mode to view the effect of each choice. CICS-WINDOWS will send an extended datastream without regard to the TCT specification, which can result in a PROG471 error if the terminal will not support extended attributes. If the PROG471 error occurs, press the Window key to exit window mode and change the option back to NO.

GRAPHICS ESCAPE

If this terminal supports graphics escape sequences, this field may be used to specify whether graphics escape characters are to be used to display the window border as a solid line around the window. Valid entries are:

YES Use graphics escape characters to display the window borders.

NO Do not use graphics escape characters.

[Note]: For graphics escape borders to be used permanently, the CICS TCT definition should specify PS as a FEATURE and this field must be coded in the User Profile. If the TCT does not have this feature coded, but you believe the terminal will support graphics escape characters, you can set this option while in window mode to view the effect. CICS-WINDOWS will use graphics escape sequences for the window borders, which can result in a erroneously displayed window borders or a PROG471 error if the terminal does not support graphics escape sequences.

HELP KEY This is used to specify a PF or PA key to be used as the CICS-WINDOWS Help Access key. For more information on the Help Access key, see section 07 - *THE HELP-WINDOWS FEATURE*.

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

NUMBER OF SESSIONS

This is the number of sessions (virtual terminals) that are available to the operator. Note that this field can not be altered. Valid values displayed are:

number A number from 2 to 9, inclusive.

NUMBER OF WINDOWS

This is the number of windows that this operator has available. Note that this field can not be altered; for standard and variable window modes, the operator may change this at any time for his terminal with the WNDO,WIN=x command. Valid values displayed are:

number A number from 2 to 9, inclusive. Note that with STANDARD or VARIABLE window modes, 4 windows is the maximum value.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PROFILE ID This is the ID of the profile that was selected for this terminal or user as specified in the Auto-Init table. Note that the Profile ID can not be changed.

SCROLL DOWN

This field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLL LEFT

This field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 10 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLL RIGHT

This field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLL UP This field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT* KEY in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLLING COLUMNS

This is the number of columns to scroll left or right when the Left or Right command is issued (or Scrolling Key is pressed).

SCROLLING ROWS

This is the number of rows to Scroll up or down when the Up or Down command is issued (or a Scrolling Key is pressed).

SESSIONS DIRECT KEYS

Direct session toggle keys. If this feature is desired, code a PF or PA key for each virtual terminal present on this physical terminal. The keys correspond one for one with each virtual terminal. That is, the first key coded will transfer control directly to virtual terminal 1, the second to virtual terminal 2, etc.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

You may omit one or more keys in the list by skipping that position. If this is done, the virtual terminal corresponding to that key position will not have a direct key assigned to it.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*). The key selected must not be the same as any other designated key.

See the subject entitled *DIRECT SESSION KEY OPERATION* for an explanation of the use of Direct Session keys.

SESSIONS PSEUDO IDS

These fields display the pseudo terminal IDs that are in use for this terminal. These fields cannot be altered.

SWITCH BACKWARD KEY

This is the PF or PA key to be used in window mode to move "backward" from the active window to the next lower window number. Each time the Switch-Backward key is pressed, control moves from the current window to the previous one in sequence until window 1 is reached, at which time control moves to window number (NUMBER OF WINDOWS).

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

SWITCH FORWARD KEY

This is the PF or PA key to be used in window mode to move "forward" from the active window to the next higher window number. Each time the Switch-Forward key is pressed, control moves from the current window to the next one in sequence until window number (NUMBER OF WINDOWS+1) is reached, at which time control moves to window number one. Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

TOGGLE BACKWARD KEY

This is the PF or PA key to be used to move "backward" from one virtual terminal to the next lower terminal number. Each time the Toggle-Backward key is pressed, control moves from the current virtual terminal to the previous one in sequence until terminal number 1 is reached, at which time control moves to virtual terminal number (NUMBER OF SESSIONS).

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

TOGGLE FORWARD KEY

This is the PF or PA key to be used to move "forward" from one virtual terminal to the next higher terminal number. Each time the Toggle-Forward key is pressed, control moves from the current virtual terminal to the next one in sequence until terminal number (NUMBER OF SESSIONS+1) is reached, at which time control moves to virtual terminal number one. Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WINDOW CONFIGURATION

This is the current window configuration that an operator with this profile will have. For standard and variable window modes, the operator may change this at any time for his terminal with the WNDO,WIN=2x, where the x is an "H" or "V". This only applies when the operators number of windows is "2". Note that this field can not be altered. Valid values displayed are:

HORIZONTAL

When the number of windows is "2", the screen will be split into horizontal windows. This applies to standard and variable window modes only.

VERTICAL When the number of windows is "2", the screen will be split into vertical windows. This applies to standard and variable window modes only.

POPUP This specifies that popup windows are in use.

WINDOW KEY

This field is used to designate a PF or PA key to be used for entering and exiting window mode. Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

WINDOWS COLOR/HILITE

If EXTENDED ATTRIBUTES is coded 'YES', 'COLOR' or 'HIGHLIGHT', this specifies the type of border that is to be used around the window when in window mode.

Two fields are present. The first field is the color of the window border. Valid entries are:

(BLU)e, (RED), (PIN)k, (GRE)en, (TUR)quiose, (YEL)low, (WHI)te

The first three characters of the color that is desired.

The second field describes the type of highlighting to be used for the window. Valid entries are:

RB The window borders are to be displayed as reverse video.

RW While the window is recessive, the entire window will be displayed as reverse video. If extended highlighting is not in effect, the text in the window will display as the color of the window borders.

IN The window borders are not to display (invisible).

NO The window borders are to remain normal.

WINDOWS DISPOSITION

These fields determine how a window is displayed while it is not the dominant window. Note that DISPOSITION is only valid for POPUP window mode. For more information on window disposition, see *OPERATION WITH POPUP WINDOWS* in section 04 - *USING THE WINDOWING FEATURE*.

FULL The window will display as a full screen in the background.

HIDE The window will not appear on the screen.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ICON The window will appear at the bottom of the screen as an icon, showing the transaction ID of the transaction that is currently active in that window.

NORMAL
The window will remain in place and may be overlapped by the dominant window.

WINDOWS NUMBER ROWS/COLS

These fields are used to specify the size of each window.

To use this feature, code the number of rows and columns that the window is to span.

[Note]: START ROW/COL and NUMBER ROWS/COLS fields contain the current positions of each window and may be altered to change the size and shape of each window. Upon initiation of CICS-WINDOWS on this terminal, these fields are assigned values depending on the window mode, window configuration and number of windows. When the window positions and/or sizes are changed through Window Adjustment commands or Pull-Down Menu Operation, the settings will automatically reflect in these fields.

WINDOWS START ROW/COL

These fields are used to specify the start position of the upper left corner of each window.

To use this feature, code the row and column position relative to "1". For more information see the note below WINDOWS NUMBER ROWS/COLS, below.

If a Full-screen window is desired, the WINDOW START ROW field for that window may be coded with an 'F'. Upon pressing ENTER, the field will display FULL SCREEN. To revert back to a regular window, enter a zero in the field.

WINDOWS SWITCH KEYS
If this feature is desired, code a PF or PA key for each window, as desired. The keys correspond one for one with each window. That is, the first key coded will perform a Window Switch command to window 1, making it the Dominant window, the second to window 2, etc. You may code as many keys as the available number of windows.

You may omit one or more keys in the list by skipping that field. If this is done, the window corresponding to that key position will not have a Window Switch key assigned to it.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*). The key selected must not be the same as any other designated key.

ENTERING COMMANDS AT THE USER CONFIGURATION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the configuration.

PF1 HELP
Display a Help window. If the cursor is positioned in an unprotected field, the help will pertain to that field, otherwise the help will pertain to the WNDO display.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PF3 EXIT Return to CICS.

PF4 SAVE Save the current settings of this User Configuration in the profile specified in the PROFILE field. This will permanently alter the profile, and affect all other users of this profile. Note that since multiple users may share a profile, this profile may be protected against the SAVE function. For more information on the User Profile, see THE USER PROFILE TABLE in section 11 - CUSTOMIZATION.

PF5 KEYS Display the User Function Key Assignment screen. (This is explained in more detail below, in *THE USER FUNCTION KEY ASSIGNMENT DISPLAY*).

THE USER FUNCTION KEY ASSIGNMENT DISPLAY

The User Function Key Assignment display allows an operator to assign nearly any valid CICS-WINDOWS command to a PF key. This display is accessed by pressing PF5 from the User Configuration display, or selecting the KEYS pull-down menu from the action bar.

FUNCTION KEY ASSIGNMENTS

The current settings of keys and commands will display in a popup window. You can change, add or erase any field. The following is a list of valid assignments along with a brief description. The list has been broken down by type of command.

TOGGLE AND WINDOW SWITCH COMMANDS

RESUME	Return to a menu which has been exited with the ESC command.
TB	Toggle-Backward
TF	Toggle-Forward
Tx	Toggle directly to session x.
WSB	Window-Switch Backward
WSF	Window-Switch Forward
WSx	Switch directly to window x.
WINDOW	Window key for entering and exiting window mode.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WINDOW MODE COMMANDS

ALTER	Alter the size of the window. (Move the bottom right corner of the window to the position of the cursor when this key is pressed.
CLEAR	Clear this window.
DOWN	Scroll window down.
FULL	Set disposition (of the current active window) to FULL.
HELP	Specify as HELP key.
HIDE	Set disposition (of the current active window) to HIDE.
ICON	Set disposition (of the current active window) to ICON.
KEYS	Display the PF key assignments.
LEFT	Scroll window left.
MENU	Display Control Window.
MOVE	Move the top left corner of the active window to the position of the cursor when this key is pressed.
RIGHT	Scroll window right.
ST	Display the toggle function window.
SW	Display the window switch function window.
UP	Scroll window up.
WDOWN	Move the window down.
WLEFT	Move the window left.
WNARROW	Make the window narrower.
WRIGHT	Move the window right.
WSHORT	Make the window shorter.
WTALL	Make the window taller.
WUP	Move the window up
WWIDE	Make the window wider.
XPAND	Expand the window to full screen mode.

CUT AND PASTE COMMANDS

COPY	Copy the field at the position of the cursor.
PASTE	Paste the clipboard at the position of the cursor.
STACK	Copy the field at the position of the cursor and append to the bottom of the clipboard.
UNPROTECT	Unprotect all fields on the screen.

CHANGING THE NUMBER OF WINDOWS

If you are using Standard or Variable window modes, you may change the number of windows (and the two-window configuration) without terminating CICS-WINDOWS by entering the command:

WNDO,WIN=x

where x is the number and configuration of the desired windows (2H, 2V, 3 or 4).

[Note]: This command is not applicable to popup windows.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TERMINATING CICS-WINDOWS

CICS-WINDOWS may be terminated in either of two ways:

1). Enter WNDO,OFF. This completely terminates CICS-WINDOWS at this terminal.

2). Do a CICS SIGN-OFF in any virtual terminal. When this happens, the following message will appear:

DO YOU WANT TO TERMINATE CICS-WINDOWS?

The default response is yes ("Y"). To terminate CICS-WINDOWS on this terminal, simply press ENTER. However, if you do not wish to terminate CICS-WINDOWS, enter a no ("N"). This will allow you to sign off CICS while leaving all virtual terminals intact.

[Note]: If the REQUIRE WNDO,OFF AT LOGOFF option has been defined in the User Option Table (see section 11 - *CUSTOMIZATION*), the following message will be displayed:

CICS-WINDOWS MUST BE TERMINATED TO LOG-OFF.

If this message appears, the terminal user must terminate CICS-WINDOWS, prior to CICS sign off.

[Note]: If FORCE PURGE AT SIGNON/SIGNOFF has been defined, CICS-WINDOWS will be automatically terminated when a signoff occurs.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE CICS-WINDOWS SYSTEM DISPLAY

The System Display provides a means of displaying the current status of CICS-WINDOWS users in the system environment. The display is invoked by entering the command:

WNDO,SYS

The display appears generally as follows:

```
WNDO,SYS
                        CICS-WINDOWS SYSTEM DISPLAY

TERM  PRO-  NO  PSEUDO LOGICAL TERMINALS ...              CURRENT USERS=00010
 ID   FILE  WINS  IDS   (1)   (2)   (3)   (4)   (5)   (6)   (7)   (8)   (9)
----  ---   ----  ---   ----  ----  ----  ----  ----  ----  ----  ----  ----
D30C  ****    2    Y    DATE  PNUM              COMP  ORDS*
D305  FRED    2    Y    DATE  PNUM* NOTE        COMP  BILL
D31A  ****    2    Y    DATE  PNUM* NEXT        COMP
D313  JOE     2    Y    DATE  NEXT  NOTE  ORDR  COMP*
V001  DON     2    Y    DATE  NEXT        ORDR  COMP  ORDS*
V007  JIM     2    Y    COMP        NOTE        COMP*
V010  FELX    2    Y    DATE  NEXT              COMP* BILL
V013  ROBT    2    Y    DATE  PNUM*
V015  LARR    2    Y    DATE  NEXT  NEXT  NEXT  COMP  WNDO*
V503  JAN     2    Y    TECH  PNUM* PRTQ




BRIGHT TRANSACTION CODES ARE CONVERSATIONAL
PF4=COMPRESSION STATS, PF5=TERMINAL IDS, TAB AND ENTER TO VIEW TXN SCREEN
```

FIELDS OF THE SYSTEM DISPLAY

The following information is displayed for every terminal in the system that is currently using CICS-WINDOWS:

1). Physical Terminal ID.
2). Profile ID in use for each terminal.
3). Number of windows in use.
4). Whether Pseudo Terminal IDs are in use or not.
5). The transaction code of the current transaction activated in each virtual terminal.
6). Compression statistics for each terminal and the entire system.

An asterisk (*) following the transaction code indicates the virtual terminal at which the operator is currently working.

If a transaction code is high intensity, this transaction is in a conversational state.

Information on up to 16 terminals may be displayed on one screen. Pressing ENTER will continue the display with the next page of terminals.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

You may begin the display with a specific terminal, by entering the command:

WNDO,SYS,xxxx

where xxxx is the terminal ID of the desired physical terminal. If that terminal is not currently using CICS-WINDOWS, the display will start with the first terminal, just as though you had entered the command: WNDO,SYS.

DISPLAYING THE PSEUDO TERMINAL IDS

When the System Display is on the screen as described above, you may press PF5, which will cause the Active Transaction codes displayed for each virtual terminal to change to the Pseudo Terminal ID that is assigned to each Virtual Terminal. Pressing ENTER will change the display back to transaction codes.

DISPLAYING DATA COMPRESSION STATISTICS

Pressing PF4 with the WNDO,SYS command will produce another display, the Data Compression Statistics display. If Data Compression has been activated (see section 11 - *CUSTOMIZATION*), this display will show you the extent of optimization that is being performed in your environment.

Optimization totals for the entire system are displayed on the first line followed by individual terminal statistics for each terminal. The display contains the following fields:

BYTES SAVED
: This is the total number of data bytes saved by using compression.

BYTES SAVED PER OUTPUT
: This is the average number of data bytes per screen display that have been saved by using data compression.

As with the other forms of the System Display, if the display is full, the next terminal to be displayed is displayed in the upper-left following the WNDO,SYS. Pressing PF4 will continue the display with the next terminal. Note that the system totals will always display on the first line.

At this point, you may press ENTER to return to the normal System Display showing CICS-WINDOWS activity, or press PF5 for the Pseudo Terminal ID display, or clear the screen and proceed with any terminal activity.

[Note]: If the DATA STREAM COMPRESSION operand of the User Option table (see section 11 - *CUSTOMIZATION*) is coded YES, rather than ALL, the totals on the SYS line only pertain to CICS-WINDOWS users.

COMPRESSED BYTE COUNT
: This is the sum of all data bytes actually sent to the terminals, after applying compression logic.

NO. OF OUTPUTS
: This is the number of outputs that have been sent to the terminal(s).

OUTPUT BYTE COUNT

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

This field is the sum of all data bytes that the programs have sent to the terminals, prior to applying any compression logic.

PERCENT This is the optimization percentage, which is arrived at by dividing BYTES SAVED by OUTPUT BYTE COUNT. The percentage is carried to one decimal place.

TERM ID

On the first line, this will be the word 'SYS', which indicates that the totals on this line are the sum of activity for all terminals in the system, whether the terminal is using CICS-WINDOWS or not. (see Note 1, below).

Subsequent lines contain the terminal IDs of active CICS-WINDOWS users. The totals on these lines apply to this terminal only.

VIEWING ANOTHER TERMINAL SESSION

The WNDO,SYS display screen can also be used on your terminal to view the screen display of another physical terminal. This can be a useful feature for problem analysis, operator training, etc.

In order to use this feature, the terminal to be viewed must be using CICS-WINDOWS. To perform a terminal session view, do the following:

1). From a clear screen, enter the WNDO,SYS command, to display the CICS-WINDOWS System Display.

2). Locate the terminal to be viewed (page forward if necessary).

3). Use the TAB key to position the cursor on the first character of the transaction code displayed in the virtual terminal to be viewed.

4). Press ENTER.

Upon pressing ENTER, the screen display that was last seen in the designated virtual session for that terminal will be displayed on your terminal. You can not respond to the screen. That is, the application program is not active on your terminal, you are simply looking at the terminal display.

When you press ENTER again, you will be returned to the beginning of the WNDO,SYS display.

[Note]: If the virtual terminal where you position the cursor has an asterisk (*) by it, it means that this is the session which the operator at that terminal is currently using. Normally, the display that you see upon pressing ENTER is not what the operator is currently looking at. What you are seeing is the screen that was saved when the operator last pressed the toggle or window key.

However, if the READ BUFFER option is not in effect for that terminal (see section 11 - *CUSTOMIZATION*), you will be viewing the same screen that is currently displayed on the virtual terminal.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USING THE CONTROL CHARACTER TO PERFORM COMMANDS

The control character can be used to perform functions such as Cut & Paste or toggling to a different session. It is keyed in an unprotected field of any screen along with the mnemonic for the function to be performed.

The Control Character is the tilde sign (~) by default. However, this is a customization option that may have been changed at your installation. For more information, see *THE USER OPTIONS TABLE* in section 11 - *CUSTOMIZATION*.

To use, simply tab to an unprotected field, key the control character along with the function to be performed, and press ENTER. For instance, to Toggle Forward to the next session, enter ~TF.

It can be used only if CICS-WINDOWS is active on the terminal. The control character function will not disturb the data that was already present in the field in which it was keyed.

See APPENDIX A - *SUMMARY OF CICS-WINDOWS COMMANDS* for a list of the commands that can be used in conjunction with the control character.

[Note] The Control Character is the tilde sign (~) by default. However, this is a customization option that may have been changed at your installation. For more information, see *THE USER OPTIONS TABLE* in section 11 - *CUSTOMIZATION*.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

OPERATION OF CICS-WINDOWS HELP

CICS-WINDOWS provides an on-line Help facility for quickly accessing operator instructions. Although it is not meant to replace this Reference Guide, the Help facility is quite extensive.

The main types of help displays are as follows:

Field Level Help

This is specific help that applies to one field of a CICS-WINDOWS display. This type of help is accessed by tabbing the cursor to the field for which help is desired, then pressing the help key.

Screen Level Help

This is general help that applies to an entire CICS-WINDOWS display. This type of help is accessed by moving the cursor out of any field (such as to the title of the screen), then pressing the help key.

Nested Help

This type appears as a help menu while already in help mode. It is accessed by tabbing to the desired option, then pressing ENTER.

Help Index

The Help Index is a type of Nested Help that provides detailed help for functions such as Window Mode Operation and Cut and Paste Procedures, and offers a basic description of functions such as Customization. It is accessed from an action bar choice or by issuing the **~H** or **WNDO,HELP** commands (explained below).

As briefly explained above, several methods may be used to access help. The method used will determine the type of help.

If a CICS-WINDOWS screen (such as the User Configuration Screen) is displayed, you may tab the cursor to any field in question, and press the help key (for field level help), or you may use the action bar and select any of the help choices on the pull-down menu.

If you are wanting help on a CICS-WINDOWS function and CICS-WINDOWS is active on the terminal, you may key the control character (see note below) along with the an 'H' in any field of the screen and press ENTER. The Help Index will then display. When help is exited, the **~H** will not appear in the field in which it was keyed. Or you may issue the **WNDO,HELP** command from a clear screen.

All types of help may be exited by pressing PF3 or the CLEAR key. While in nested help, the help key may be pressed to return from the current help display to the previous level.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE CICS-WINDOWS CONTROL WINDOW

The Control Window allows operators to perform functions such as Window Switching and Direct Toggling without the need to memorize commands and/or PF keys. The Control Window can be displayed by keying the Control Character (see note below), then the letters "MENU", anywhere on the screen, then pressing ENTER. Thus, if the control character is the tilde sign (~), the command **~MENU** (or simply **~M**) typed anywhere on the screen would invoke the Control Window, provided CICS-WINDOWS is active on the terminal. In addition, you may also have a PF or PA key that will invoke the Control Window, and the menu may be invoked from the action bar of a popup window.

[Note] The Control Character is the tilde sign (~) by default. However, this is a customization option that may have been changed at your installation. For more information, see *THE USER OPTIONS TABLE* in section 11 - *CUSTOMIZATION*.

When invoked, the Control Window is displayed in a popup window. The functions that are available are detailed below. Note that since the Control Window is created by the CICS-WINDOWS Menu generation facility, the menu may have been further customized at your installation.

DISPLAY SESSION SELECTION MENU

This may be used as an alternative to the toggle keys or sessions direct keys. When this option is selected, a popup window will appear showing information about each session such as window number, virtual terminal number and active transactions. To perform the toggle, tab the cursor to the desired session number and press ENTER.

DISPLAY WINDOW SELECTION MENU

This may be used as an alternative to window switch keys. When this option is selected, a popup window will appear showing information about each window, such as session number, virtual terminal number and active transactions. To perform the window switch, tab the cursor to the desired window number and press ENTER.

ENTER/EXIT WINDOW MODE

This has the same effect as pressing the Window Key, but may be used instead, if a Window Key is not available.

DISPLAY FUNCTION KEY ASSIGNMENTS

This option is the same as pressing the KEYS key while viewing the User Configuration Display. For more information, see *THE USER FUNCTION KEY ASSIGNMENT DISPLAY*, earlier in this section.

DISPLAY MESSAGES

This option can be used to display a message that has been broadcast to this terminal. For more information, see section 09 - MESSAGE BROADCASTING.

EXIT CONTROL WINDOW

To exit the Control Window, select this option.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

OPERATION OF ACTION BARS

Action Bars are located throughout CICS-WINDOWS and operate according to CUA standards. An Action Bar is a list of available actions that appears at the top of a display. It is much like a function key area, except the action bar does not list the actions directly. Instead, the actions are grouped together on pull-down menus, and the group names are what is listed in the action bar.

The remainder of this topic describes the use of action bars in general terms. This discussion does not attempt to detail the actual contents of every action bar.

INITIATING AN ACTION THROUGH A PULL-DOWN MENU

The technique for initiating actions with an action bar, is to pull down the desired menu, then from the pull-down menu, select the action to be performed. In order to pull down a menu, there are two methods:

- Tab the cursor to the name of the menu, then press ENTER.
- Key the first letter of the name of the menu in the action bar entry field, then press ENTER.

  Note that some action bars may have conflicting menu mnemonic selections. In this case, a menu name will have a letter enclosed in parenthesis. This is the letter to be keyed in the entry field.

Upon selecting a menu, a pull-down menu will drop down from the action bar. Now you may select an action with any of the following methods:

- Tab the cursor to the name of the action, then press ENTER.
- Key the selection number of the action to be performed in the menu entry field, then press ENTER.
- In some instances, an action may be initiated by pressing a function key. For these actions, the function key will be listed in the menu along with the action, and may be pressed to select the desired action.

If you do not wish to invoke an action while a pull-down menu is displayed, you may either select a different menu, select the RESUME function, or press CLEAR to remove the pull-down menu.

INITIATING AN ACTION THROUGH THE FAST PATH METHOD

As an alternate technique for initiating an action with the action bar, the fast path method may be used. For this method, simply key the menu mnemonic and the selection number of the action to be performed in the action bar entry field, then press ENTER.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USING APPLICATIONS WITH CICS-WINDOWS ACTIVE

This section will discuss how to operate applications while using **CICS-WINDOWS.** Examples will be given to aid the user.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 6. USING THE CUT AND PASTE FEATURE

Cut and paste, or more properly, copy and paste, is a feature of CICS-WINDOWS which allows you to copy selected data from one application display and insert this data into another application display, or elsewhere in the same display.

Cut and paste has many operational benefits and uses, some of which are:

- Avoid duplicate keying of data by copying redundant information from another application display.
- Capture partial or entire screen displays and insert them into an editor or word processor for text manipulation or printing.
- Propagate fields of data into other fields.

Cut and paste operates by command entry and use of the ENTER key, but the function may optionally be assigned to PF keys in the User Function Key Assignment display.

You can copy fields in full-screen mode or in window mode.

Copied fields are held in a scratch-pad area until a successive copy is performed. There can be any number of transaction entries between a COPY command and a PASTE command.

Once the PASTE command is completed, the scratch pad area containing the copied field(s) is retained. This allows one to paste the same information on multiple displays without the necessity of performing multiple copies.

COMMANDS USED IN CUT AND PASTE

Two commands are used to accomplish a cut and paste operation. They are COPY ("C") and PASTE ("P").

If you attempt to PASTE without first performing a COPY, the following message will display:

NOTHING TO PASTE

One additional command may be used when the field to be copied is a protected field. This is the 'unprotect' (UNP) command. UNP will set all attributes on the screen to unprotected, thereby allowing the copy command to be entered in any field.

Note that the transaction in the session will not operate following an unprotect command. The next time ENTER or a function key is pressed, the screen will restore with the correct attribute structure and the transaction may continue.

All cut and paste commands are entered by typing over some data of the application display, though after the function is performed, the previous data in the display will not be disturbed. The command must be preceded by the command character. The default command character is a tilde (~), but it may be changed in the customization table.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE COPY COMMAND

The COPY command is entered in a data field of the application display as the command character followed by the COPY command code. Thus, if the default values have not been changed in the customization table, the command would be ... ~C (tilde, 'C').

For individual field copies, the data copied begins with the character that occupied the space where the command character is keyed, then continues to the end of the field (until the next attribute is found).

For line copies (see *TYPES OF CUT AND PASTE OPERATIONS*), the data copied begins with the first character of the line, or row, where the command was entered. For line copies, it does not matter where in the line the COPY command is placed.

For full-screen copies, the COPY command can be placed anywhere on the screen. The data copied begins with the first character on the screen (row 1, column 1) and continues to the end of the screen.

THE STACK COMMAND

The STACK command may be used to append copied information to the scratch pad area without deleting previously copied data.

The STACK command operates exactly like the COPY command (with the above mentioned exception). The STACK command is entered in a data field of the application display as the command character followed by the STACK command code. Thus, the command would be ... ~S (tilde, 'S').

THE PASTE COMMAND

The PASTE command is entered in a data field of the application display as the command character followed by the PASTE command code. Thus, if the default values have not been changed in the customization table, the command would be ... ~P (tilde, 'P').

For individual field pastes, the data pasted begins at the location where the command character is keyed, then continues to the end of the field (until the next attribute is found). If the data that was copied exceeds the length of the paste field, it is truncated on the right. If the copied data is shorter than the paste field, the remainder of the paste field is padded with spaces.

For line copies (see *TYPES OF CUT AND PASTE OPERATIONS*), the data pasted begins at the location where the command character is keyed, then continues to the end of the line. Consecutive characters of the copied line (starting with the first character) will be pasted into all unprotected fields on the line (unless a field count value was entered with the PASTE command). If the copied line exceeds the length of the paste line, excess characters of the copied line are truncated on the right. Consecutive lines of the copied data, starting with the first character of each copied line, will be pasted into consecutive lines of the screen, each pasted line beginning at the same location on its respective line where the command character was placed on the line containing the PASTE command. If the number of lines copied exceeds the number of lines available to be pasted, excess copied lines are dropped.

For full-screen copies, the data pasted begins at the location where the command character is keyed, then continues to the end of the screen. If the data copied exceeds the paste screen size, excess characters of the copied screen are dropped.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

[Note]: For an understanding of individual field copies, line copies and full-screen copies, see *TYPES OF CUT AND PASTE OPERATIONS*, below.

THE UNPROTECT COMMAND

The unprotect command can be entered at any location of the application display. It is entered as the command character followed by 'UNP'. Thus, if the default command character is in use, the command would be ~UNP.

When the UNP command is entered, all field attributes on the screen are set to unprotect. The intensity value remains unchanged. This allow you to enter a COPY command in any data field of the screen, even title and text fields.

Following an unprotect command, the transaction in this session is temporarily suspended. You cannot operate the transaction until you press ENTER or some other function key with no command character on the screen. At that time, the original screen is refreshed with the correct attribute structure, whereupon the transaction may be continued.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TYPES OF CUT AND PASTE OPERATIONS

The COPY and PASTE commands can accomplish any of the following functions:

1). Copy a single field from a display and paste it elsewhere maintaining the same attribute structure.

2). Copy multiple fields from a display and paste them elsewhere maintaining the same attribute structure.

3). Paste the same field into multiple other fields (field propagation).

4). Copy one or more lines (copying all fields on the line) of a display, clear all field attributes and paste the line(s) elsewhere.

5). Copy an entire screen display, maintain the same attribute structure and paste the entire screen over another screen or a blank screen.

6). Copy an entire screen display, clear all attributes and paste the entire screen over another screen or a blank screen.

Each of these functions differ in their usage and operation and it is important to understand each in order to know when to use the correct type of copy and paste to accomplish your purpose.

SINGLE-FIELD CUT/PASTE

To copy a single field, type the COPY command (~C if the default command character is in use) starting at the first position of the field to be copied. This does not have to be the first byte of the field. All data in the field from the location of the command character to the end of the field will be copied and stored in a scratch-pad area.

Now you may toggle to another session, switch to another window (if in window mode), or move to another field of the same application display, then type the PASTE command (~P if the default command character is in use) starting at the first position of the field where the data is to be inserted. This does not have to be the first byte of the field. All data in the field from the location of the command character to the end of the field will be replaced by the data of the field that was copied. If the data that was copied exceeds the length of the paste field, it is truncated on the right. If the copied data is shorter than the paste field, the remainder of the paste field is padded with spaces.

When a field is pasted, the MDT (modified data tag) for that field is set on. Thus, when ENTER or a function key is pressed, that data field will transmit to the receiving application, just as if it were keyed by the operator.

MULTIPLE-FIELD CUT/PASTE

To copy multiple fields, type the COPY command starting at the first position of the first field to be copied. This does not have to be the first byte of the field. Now place the command character in all subsequent fields to be copied. It is only necessary to have the COPY command on the first field to be copied. You can enter it is subsequent fields if desired, but just the presence of the

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

command character is sufficient. All data in each field from the location of the command character to the end of the field will be copied and stored in a scratch-pad area.

Now you may toggle to another session, switch to another window (if in window mode), or move to another field of the same application display, then type the PASTE command starting at the first position
of the first field where the data is to be inserted. This does not have to be the first byte of the field. Now place the command character in all subsequent fields to be pasted. It is only necessary to have the PASTE command on the first field to be pasted. You can enter it is subsequent fields if desired, but just the presence of the command character is sufficient. All data in the field from the location of the command character to the end of the field will be replaced by the data of the corresponding field that was copied. If there are less paste fields than the number of copied fields, the excess copied data is dropped. If there are more paste fields than the number of copied fields, the last copied field is propagated into the excess paste fields.

For each pasted field, the MDT (modified data tag) for that field is set on. Thus, when ENTER or a function key is pressed, those data fields will transmit to the receiving application, just as if they were keyed by the operator.

FIELD PROPAGATION CUT/PASTE

Field propagation is simply duplicating a single field multiple times on the same or another screen. It is accomplished with the PASTE command, after doing a single or multiple field COPY.

If only one field was copied, you can enter a PASTE command in a receiving field, then continue to key the command character in additional fields (any number). When ENTER is pressed, the copied field will be pasted into every location where the PASTE command and the additional command characters were placed.

If multiple fields are copied, field propagation will occur if you key more PASTE locations then the total number of fields copied. In this case, the copied fields are pasted one-for-one until they are exhausted. Then, the final copied field continues to be pasted into the additional locations until all are complete.

LINE CUT/PASTE

To copy multiple lines of a screen display, type the COPY command starting anywhere on the first line to be copied. Follow the COPY command with a comma and a 2-digit number which is the number of lines to be copied. Thus, to copy 9 lines, the command would be entered as

~C,09

Starting with row 1 of this line, all data on every line from this row through the ending row will be copied and stored in a scratch-pad area. All attributes present on each copied line are set to spaces. Thus, the attribute structure of individual fields on the line is lost. The lines become, in effect, pure textual data.

Now you may toggle to another session, switch to another window (if in window mode), or move to another field of the same application display, then type the PASTE command starting at the first position on the line where the data is to be inserted.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Starting with the position on the first line where the command character was placed, each copied lines will be pasted, beginning with the first byte of data of the copied line. The paste will continue to the end of the line, pasting into unprotected fields only. Any excess data of the copied line that will not fit on the paste line is dropped. The paste then continues to the next line, starting with the same relative position on this line corresponding to the position of the command character on the first line.

The paste will continue until either the end of the paste screen is reached or the total number of lines that were copied is exhausted.

For each pasted field, the MDT (modified data tag) for that field is set on. Thus, when ENTER or a function key is pressed, those data fields will transmit to the receiving application, just as if they were keyed by the operator.

One variation of the PASTE command, when used with a line copy, is to restrict the number of unprotected fields to be pasted on the receiving lines. By following the PASTE command with a comma and a 1-digit number, the number of fields of each line to be pasted can be specified.

Thus, if the PASTE command is entered as

```
~P,1
```

it would mean that only one field in each line is to be pasted and all remaining data of the copied line is to be discarded.

The line copy and paste operation is designed to be used with full-screen editors or word processors. It allows you to copy any sort of data, such as an application screen display, and paste this data into a series of lines of a text editor, thereby capturing the application screen display and converting it into textual data that could be manipulated or printed with the editor. It could also be used to copy multiple lines from one text editor into another.

FULL SCREEN CUT/PASTE

There are two variations of the full-screen cut/paste operation which may be used. The first method will copy an entire screen display and leave the attribute structure intact. The second method will copy the entire screen and clear all attributes to spaces, thereby converting the display to pure textual data.

To copy an entire screen leaving the attribute structure intact, enter the COPY command anywhere on the screen. The command is entered as the normal COPY command followed by a comma and the word 'ALL'. Thus, if the default command character is in force, the command would be entered as

```
~C,ALL
```

All data on the screen from row 1, column 1 to the end of the screen, including all field attributes, will be copied and stored in a scratch-pad area.

Now you may toggle to another session or switch to another window (if in window mode), then type the PASTE command anywhere on the receiving screen. The receiving screen could be a blank screen, if desired.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

All data of the receiving screen is completely replaced with the copied screen, with no modification made to the attributes of the copied screen.

This function has limited usefulness. It may be possible in some cases to copy a screen into a blank screen, then press ENTER to initiate the same application in another session. This will only work, however, if the screen has the MDT set on the appropriate field(s) to activate the application (such as the transaction code, for CICS transactions). The PASTE operation will not set MDT on for any fields when used in this manner. It is possible, after pasting the screen, to key over any unprotected fields of the display, thereby turning on the MDT, and then press ENTER, causing those fields to be transmitted.

To copy an entire screen converting all attributes to spaces enter the COPY command anywhere on the screen. The command is entered as the normal COPY command followed by a comma and the word 'ALLCLR'. Thus, if the default COPY command code and command character are in force, the command would be entered as

```
~C,ALLCLR
```

All data on the screen from row 1, column 1 to the end of the screen will be copied and stored in a scratch-pad area. In the process, all attributes and nulls will be converted to spaces, thereby transforming the screen image into textual data.

Now you may toggle to another session or switch to another window (if in window mode), then type the PASTE command anywhere on the receiving screen. The receiving screen could be a blank screen, if desired.

All data of the receiving screen is completely replaced with the copied screen, and the MDT will be set on for the beginning of the screen, row 1 column 1. The pasted screen will now be formatted as one contiguous field from row 1 to the end of the screen.

This function also has limited usefulness. You cannot usually use this function to copy a screen into a text editor, since the control fields of the text editor will be destroyed in the process. You would need to use the line copy for that. It may be possible in some cases to copy a screen into some sort of screen-paint application, such as an application development system, then re-key the necessary control fields and thereby copy that screen into the application.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

EXAMPLES OF CUT/PASTE OPERATIONS

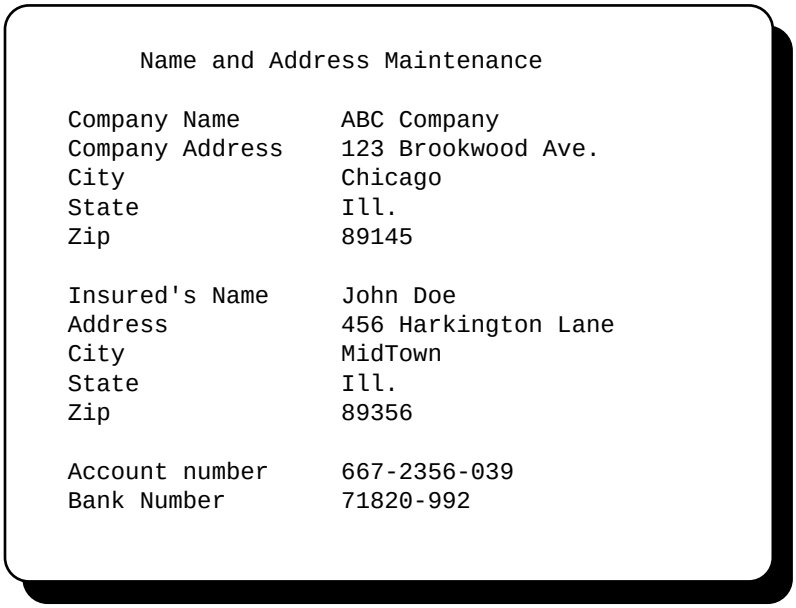
Following are examples of each of the various types of cut and paste operations. Single field copy, multiple field copy, field propagation, and line copy are illustrated.

These examples are intended to illustrate a few meaningful situations where cut and paste can be used to reduce keying time, or capture information that normally could not be captured at all without extensive keying.

As you use the cut and paste feature, you will find many applications where it is useful in your environment.

SINGLE FIELD CUT/PASTE EXAMPLE

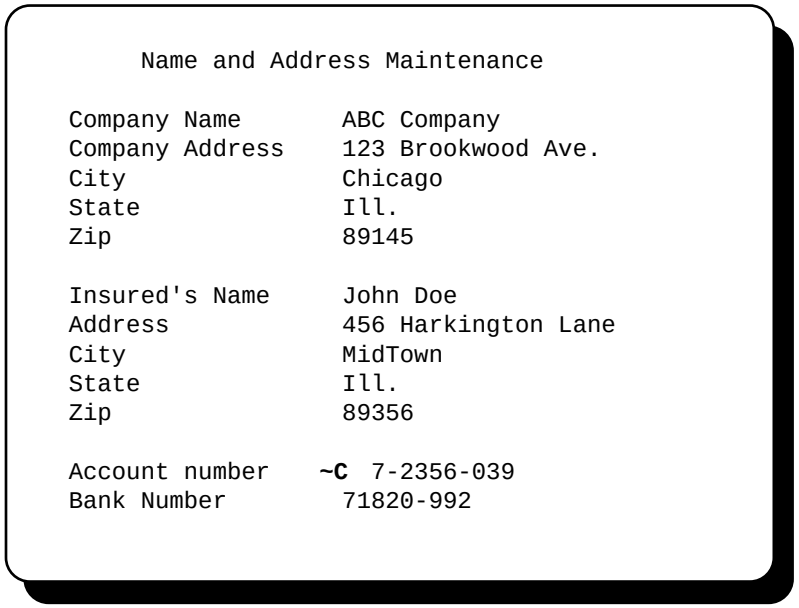
Session 1 contains the following application display:



We want to copy the account number field and paste it into another application. To do this, we move the cursor to the field and enter the COPY command, preceded by the command character.

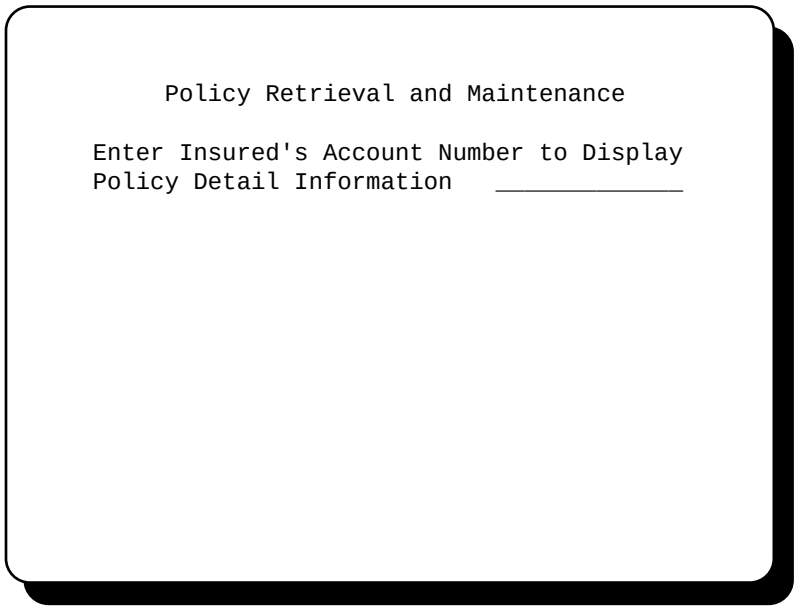
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

It will appear as follows:



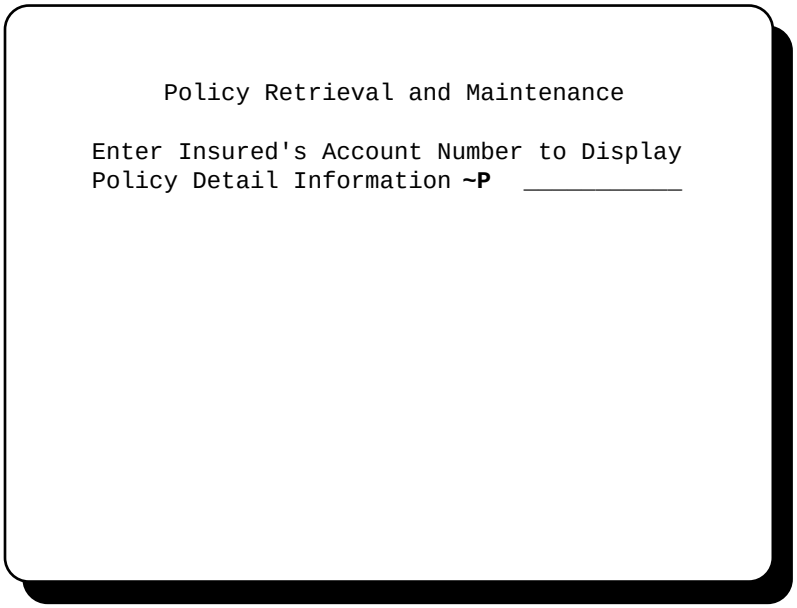
Now we press ENTER, which will re-display the original screen with the message COPY FUNCTION COMPLETED on the bottom row.

Now we toggle to session 2, where the following key-entry application display is waiting:

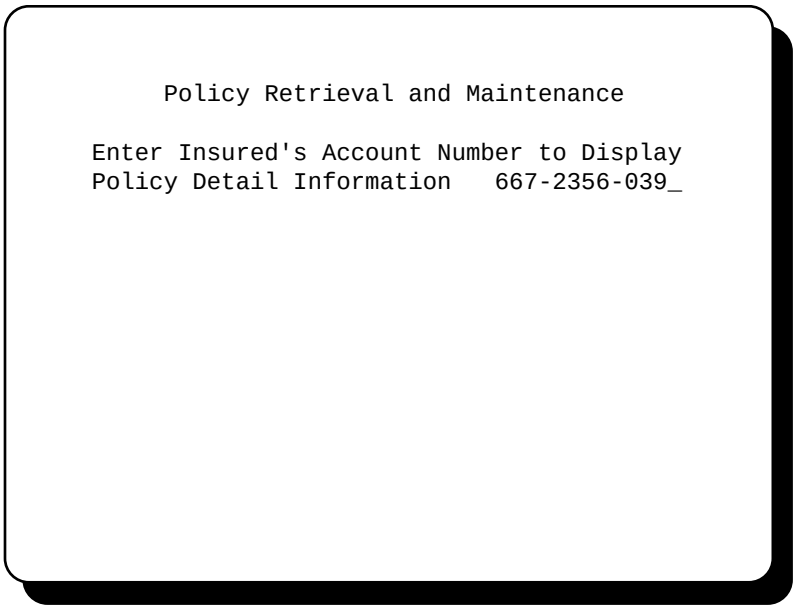


© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

We move to the entry field and key the PASTE command as follows:



Upon pressing ENTER, the copied account number will be inserted into the receiving field, whereupon the operator may press ENTER to transmit the number to the application.



This example illustrate both the saving of key strokes and the insurance of accuracy when re-keying an account number for a different application.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MULTIPLE FIELD CUT/PASTE EXAMPLE

Session 1 contains the following application display:

```
      Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name     John Doe
Address            456 Harkington Lane
City               MidTown
State              Ill.
Zip                89356

Account number     667-2356-039
Bank Number        71820-992
```

We want to copy the Insured's name and address and paste it into another input screen of the same application. To do this, we place the COPY command in the insured's name field, then key the command character in the subsequent fields of the address, as follows:

```
      Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name    ~C  hn Doe
Address           ~  56 Harkington Lane
City              ~  idTown
State             ~  ll.
Zip               ~  9356

Account number     667-2356-039
Bank Number        71820-992
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Upon pressing ENTER, we may now toggle to session 2, activate the application to do a record addition, key the company name/address, the place the PASTE command in the insured's name and the command character in the subsequent address fields, as follows:

```
     Name and Address Maintenance

Company Name       ABC Company
Company Address    123 Brookwood Ave.
City               Chicago
State              Ill.
Zip                89145

Insured's Name   ~P ____________________
Address          ~  ____________________
City             ~  ____________________
State            ~  ____
Zip              ~  ____

Account number     ___________
Bank Number        _________
```

When ENTER is pressed, the copied fields will be inserted into the corresponding screen fields, as follows:

```
          Name and Address Maintenance

     Company Name       ABC Company
     Company Address    123 Brookwood Ave.
     City               Chicago
     State              Ill.
     Zip                89145

     Insured's Name     John Doe
     Address            456 Harkington Lane
     City               MidTown
     State              Ill.
     Zip                89356

     Account number     ___________
     Bank Number        _________
```

Now the remaining two fields can be keyed and ENTER pressed to complete the transaction.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELD PROPAGATION CUT/PASTE EXAMPLE

Session 1 contains the following text editor display:

```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100        Move Work-Master-Name to Screen-Name.
0110 ____________________________________________________
0120 ____________________________________________________
0130 ____________________________________________________
0140 ____________________________________________________
0150 ____________________________________________________
0160 ____________________________________________________
```

We want to copy the last line (number 0100) and replicate it into the remaining six lines in order to avoid keying "Move Work-Master ..... to Screen" six more times.

To do this, we can do a single field copy on the text of line 0100 as follows:

```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100  ~C    Move Work-Master-Name to Screen-Name.
0110 ____________________________________________________
0120 ____________________________________________________
0130 ____________________________________________________
0140 ____________________________________________________
0150 ____________________________________________________
0160 ____________________________________________________
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Then, after pressing ENTER to complete the copy, place a PASTE command on line 0110 and place the command character on all subsequent lines, as follows:

```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100        Move Work-Master-Name to Screen-Name.
0110~P_____________________________________________
0120~ _____________________________________________
0130~ _____________________________________________
0140~ _____________________________________________
0150~ _____________________________________________
0160~ _____________________________________________
```

Now, when ENTER is pressed, line 0100 will be propagated into the remaining six lines.

```
==>
0010  Procedure Division.
0020  Start-Processing.
0030        Perform Handle-condition thru Handle-Exi
0040        Exec CICS Address CSA (CSA-BLL)  end-exe
0050        Move CSACDTA to TCA-BLL.
0060        Exec CICS receive map ('SCm0010')
0070                 into Input-Screen.
0080        Exec CICS read dataset ('MASTER')
0090                 into Work-Record.
0100        Move Work-Master-Name to Screen-Name.
0110        Move Work-Master-Name to Screen-Name.
0120        Move Work-Master-Name to Screen-Name.
0130        Move Work-Master-Name to Screen-Name.
0140        Move Work-Master-Name to Screen-Name.
0150        Move Work-Master-Name to Screen-Name.
0160        Move Work-Master-Name to Screen-Name.
```

Now we need only change the final qualifier of each field name to accomplish the six additional move statements.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELD CUT/PASTE EXAMPLE USING THE STACK COMMAND

Session 1 contains the following application display:

```
        Name and Address Maintenance

Company Name        ABC Company
Company Address     123 Brookwood Ave.
City                Chicago
State               Ill.
Zip                 89145

Insured's Name      John Doe
Address             456 Harkington Lane
City                MidTown
State               Ill.
Zip                 89356

Account number      667-2356-039
Bank Number         71820-992
```

We want to copy the Company name and address and paste it into another input screen of the same application. To do this, we place the COPY command in the Company name field, then key the command character in the subsequent fields of the address, as follows:

```
        Name and Address Maintenance

Company Name      ~C  C Company
Company Address   ~  23 Brookwood Ave.
City              ~  hicago
State             ~  ll.
Zip               ~  9145

Insured's Name      John Doe
Address             456 Harkington Lane
City                MidTown
State               Ill.
Zip                 89356

Account number      667-2356-039
Bank Number         71820-992
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Upon pressing ENTER, If we decide that we should also copy the Insured's name and address we could place the STACK command in the Insured's name field, then key the command character in the subsequent fields of the address, as follows:

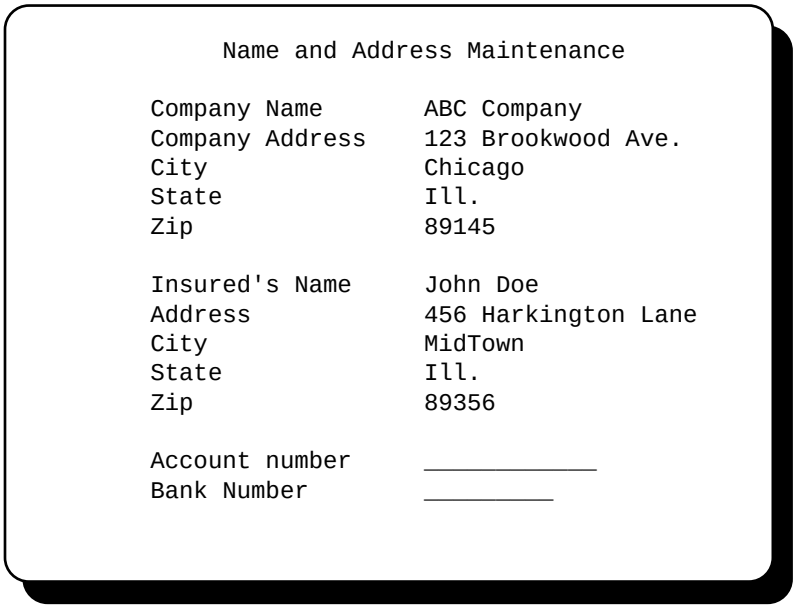
```
          Name and Address Maintenance

     Company Name        ABC Company
     Company Address     123 Brookwood Ave.
     City                Chicago
     State               Ill.
     Zip                 89145

     Insured's Name    ~S  hn Doe
     Address           ~  56 Harkington Lane
     City              ~  idTown
     State             ~  ll.
     Zip               ~  9356

     Account number      667-2356-039
     Bank Number         71820-992
```

We may now toggle to session 2, activate the application to do a record addition and the place the PASTE command in the Company name and the command character in the subsequent address fields, as follows:

```
          Name and Address Maintenance

     Company Name      ~P  _________________
     Company Address   ~  _________________
     City              ~  _________________
     State             ~  ___
     Zip               ~  ____

     Insured's Name    ~  _________________
     Address           ~  _________________
     City              ~  _________________
     State             ~  ___
     Zip               ~  ____

     Account number      ____________
     Bank Number         _________
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

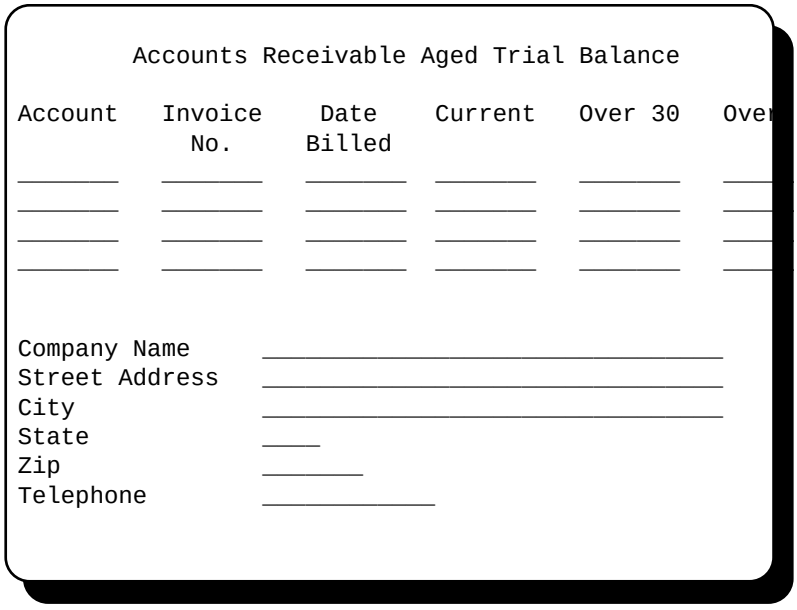
When ENTER is pressed, the copied fields will be inserted into the corresponding screen fields, as follows:



Now the remaining two fields can be keyed and ENTER pressed to complete the transaction.

LINE CUT/PASTE EXAMPLE USING THE UNP COMMAND

Session 1 contains the following application display:



We're working on a documentation project and want to capture this screen image in textual form so that it can be printed as a part of the user manual.

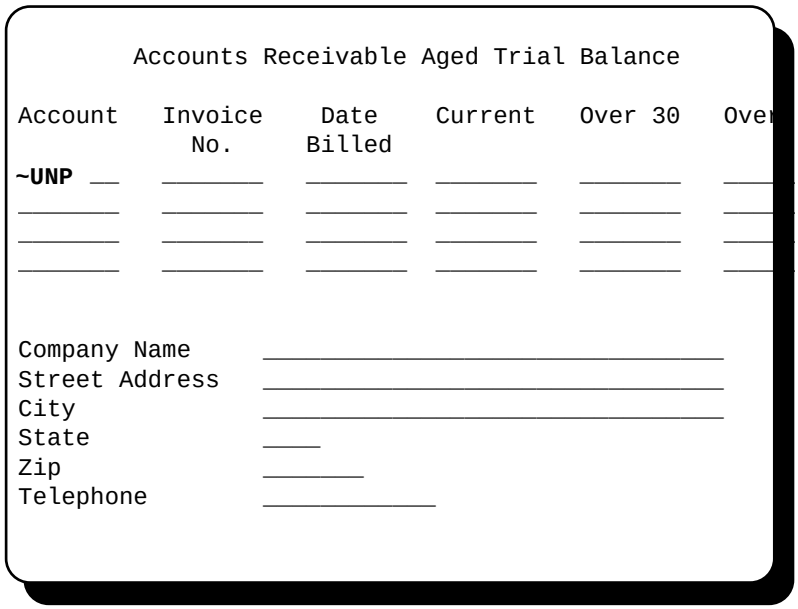
To do this, we need to use a line copy to copy 16 lines, starting with the first line, the header description. With the line copy, you must place the COPY command on the first line to be copied, then follow it with a

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

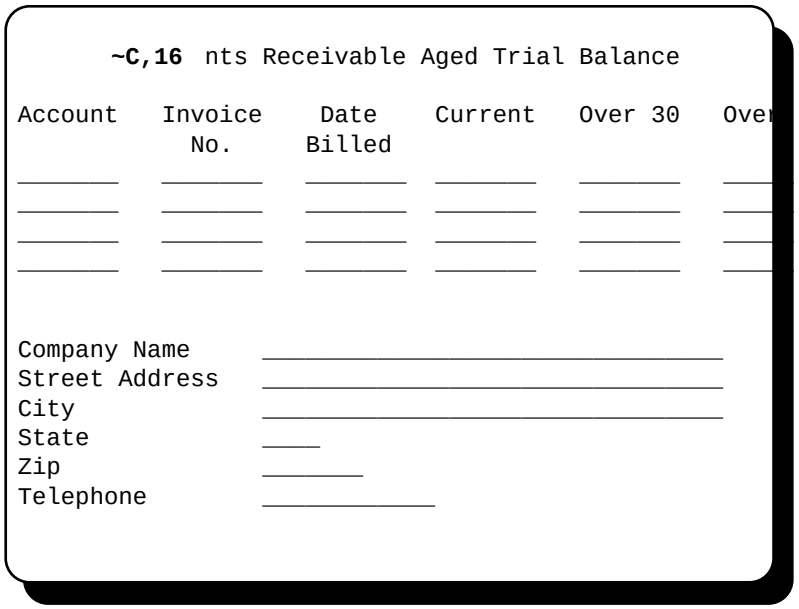
comma and the number of lines to copy. It does not matter where on the line we place the COPY command.

Since the first line is protected and we cannot key the COPY command on the line, we must first use the UNP command to unprotect this screen. This will cause all screen attributes of the display to change to unprotected, meaning that we can key over the title fields of the screen.

We do this by keying the UNP command anywhere on the screen, as follows:



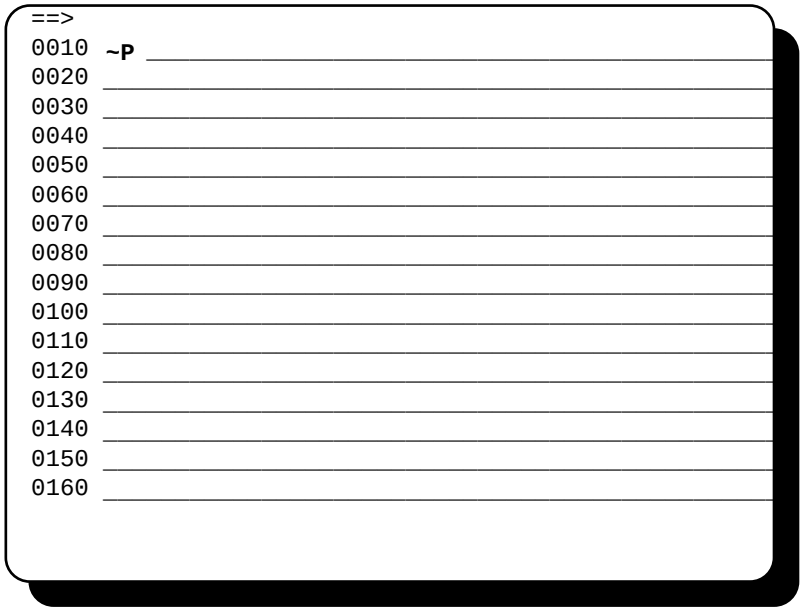
After pressing ENTER, all fields on the screen are unprotected, allowing us to key the line copy command on the header line, as follows:



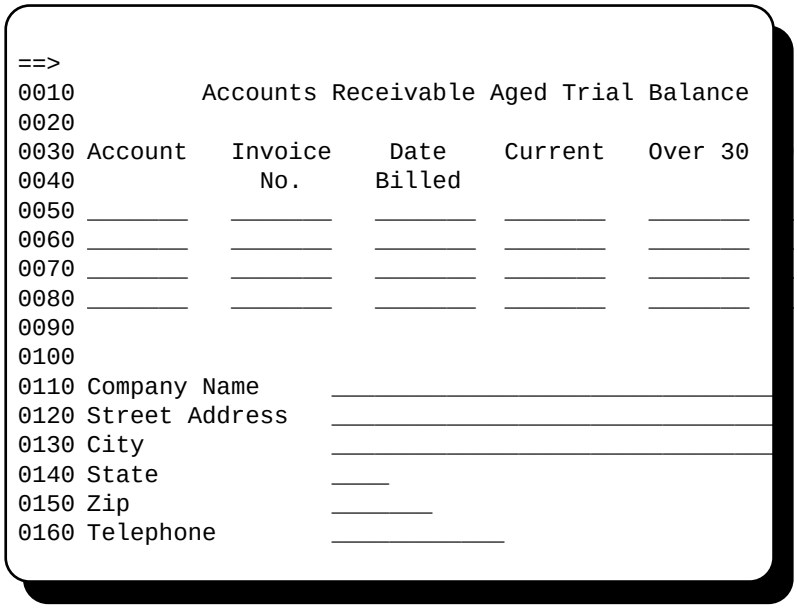
This causes 16 rows to be copied, starting with the first row of data. All attributes and nulls are translated to spaces.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Now we toggle to session 2 where we have a text editor input screen ready to receive the data. We place the PASTE command in the starting position of the first text field, as follows:



After pressing ENTER, the copied lines are pasted into 16 consecutive lines. There will be a few positions truncated off the right, since the text entry field did not start at position 1 of the line. It will appear as follows:



When we press ENTER to the text editor, each pasted line will be transmitted just as if we had keyed all of the data. The display could now be manipulated and printed as desired.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 7. THE HELP-WINDOWS FEATURE

An optionally licensed feature of CICS-WINDOWS is the HELP-WINDOWS feature, which enables users to create on-line help displays for any CICS transaction.

Help displays can be created to display either in full-screen mode or in a pop-up window on the screen. Help can be accessed at the transaction level, the screen level and/or the field level.

The process of creating help is interactive. That is, you initiate the transaction to be documented, display the desired screen, then proceed to add your help text directly to that screen through the use of the AUTO-DEFINE process.

PREPARING TO DEFINE A HELP DISPLAY

Before you can create a help display, you must have two PF keys available. These are the Help Definition (DEFINE) key and the Help Access (HELP) key.

The Define key is used in the creation of help only and is not needed by the end user who views and uses the help. The Help key is used to "test" your help displays as you create them, and by the end user to view them.

The HELP-WINDOWS feature of CICS-WINDOWS is a subset of the powerful HELP-WINDOWS package available from Unicom Systems. It is designed to support on-line help displays only, rather than full documentation systems, manual printing, etc.

Some of the features and functions of the HELP-WINDOWS features are:

- Associate help windows with a transaction, a screen or a field.
- Associate help windows with a word or phrase within other help text, thereby creating multiple levels of help.
- Display help text in a window on the screen.
- Create and use help menus, either in window mode or full screen mode, for quick access to various sections of help text.
- “Wrap” the text to fit the window, if it exceeds the window boundaries.
- Scroll forward and backward in a help display.
- No limit to the number of pages of help text in a single help definition.
- Use row/column address or associate the cursor position with text on the screen to define a help display.
- Define “cross-document” fields. Fields that appear on multiple screens or multiple transactions, always referencing a common set of help text.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

- Define help displays by invoking the user transaction screen, positioning the cursor as desired and pressing a “help definition” key.
- “Chain” sections of help text together in various ways to create multiple paths through the documentation.
- Create multi-window displays. Windows may overlap in any configuration desired. There is no limit to the number of windows on one screen.
- Create and use a table of contents for easy access to many help topics.
- “Include” modules of help text, linking them together to form a contiguous help display.
- Import text from some other file into the help-windows text file. Control statements make it easy to define and sub-define the imported document.

DEFINING THE HELP KEY

The choice of which PF key to use for help retrieval is performed at the CICS-WINDOWS User profile, as follows.

1) Enter the WAUX trancode from a clear screen.
2) Select User Profiles.
3) Locate the profile or profiles to be used, or make the same change to all profiles.
4) Enter the PF key mnemonic for the key to be used in the HELP KEY field.

[Note] For CUA compatibility, PF1 should be used as the Help key, which will not conflict with the on-line help available in CICS-WINDOWS.

DEFINING THE HELP-DEFINITION (DEFINE) KEY

The choice of which PF key to use for help definition can also be performed at the CICS-WINDOWS User profile, if desired.

1) Enter the WAUX trancode from a clear screen.
2) Select User Profiles.
3) Locate the profile or profiles to be used
4) Select the KEYS pull-down menu, then select Display Function Keys.
5) At the Function Keys display, enter the DEFINE command beside the PF key you wish to use.
6) Exit the Keys display with PF3.

[Note] This method will permanently assign this PF key as the Define key whenever this profile is in use.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

AN ALTERNATE METHOD

You can temporarily assign a Define key with the following command:

WHLP,DEFINE,PFxx

where PFxx is the mnemonic of the PF key to be used for the Define key (PF1, PF2 ... PF24). With this method, it does not matter what profile is in use. The chosen Define key will remain in force on this terminal until a WNDO,OFF is performed or WHLP,DEFINE=OFF.

ENTERING AUTO-DEFINE MODE

Before you can create a help definition, you must establish your terminal as authorized to use Auto-Define. This is performed with the command:

WHLP,DEFINE

As previously stated, you can optionally follow the command with the mnemonic of the PF key to be used as the Define key. If you are authorized to use the WHLP transaction code, you can perform Auto-Define.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CREATING A FIELD LEVEL HELP DISPLAY, A QUICK OVERVIEW

You are now ready to create and test an on-line help display. For our example, we will use a user transaction screen called HDMO. This display transaction is distributed with the CICS-WINDOWS product. It is intended to be a practice vehicle in learning to create help definitions.

STEP 1. INVOKING THE USER TRANSACTION

Enter a transaction code which will display a screen that you would like to document. For our purposes in this illustration, we will use the HDMO transaction.

Upon keying HDMO and pressing ENTER, the HDMO display will appear as follows:

```
SCREEN: CM010N                                              OPERATOR: TD
                   SAMPLE CICS APPLICATION SCREEN
               FOR HELP-WINDOWS AUTO-DEFINE TESTING


CUSTOMER NUMBER      _____
CUSTOMER NAME        ____________________________
ADDRESS LINE 1       ____________________________
ADDRESS LINE 2       ____________________________
CITY                 ____________________________
STATE __   ZIP CODE  _____


.......................  PRODUCTS INSTALLED  .......................
  PRODUCT     DATE     ORIGINAL    VENDOR     MAINTENANCE      EXPIRATION
   CODE       SOLD       COST      NUMBER       AMOUNT           DATE
   ____    _________  _________   _______    _________        _________
   ____    _________  _________   _______    _________        _________
   ____    _________  _________   _______    _________        _________
   ____    _________  _________   _______    _________        _________
   ____    _________  _________   _______    _________        _________

        USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
        REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.
```

STEP 2. POSITIONING THE CURSOR

Now position the cursor to the desired field on the screen where help is to be made available. For this exercise, let's use the CUSTOMER NAME field.

There are two methods of correlating help to a field. One is using absolute row/column cursor position. For this method, you should position the cursor to the first position of the actual data entry field.

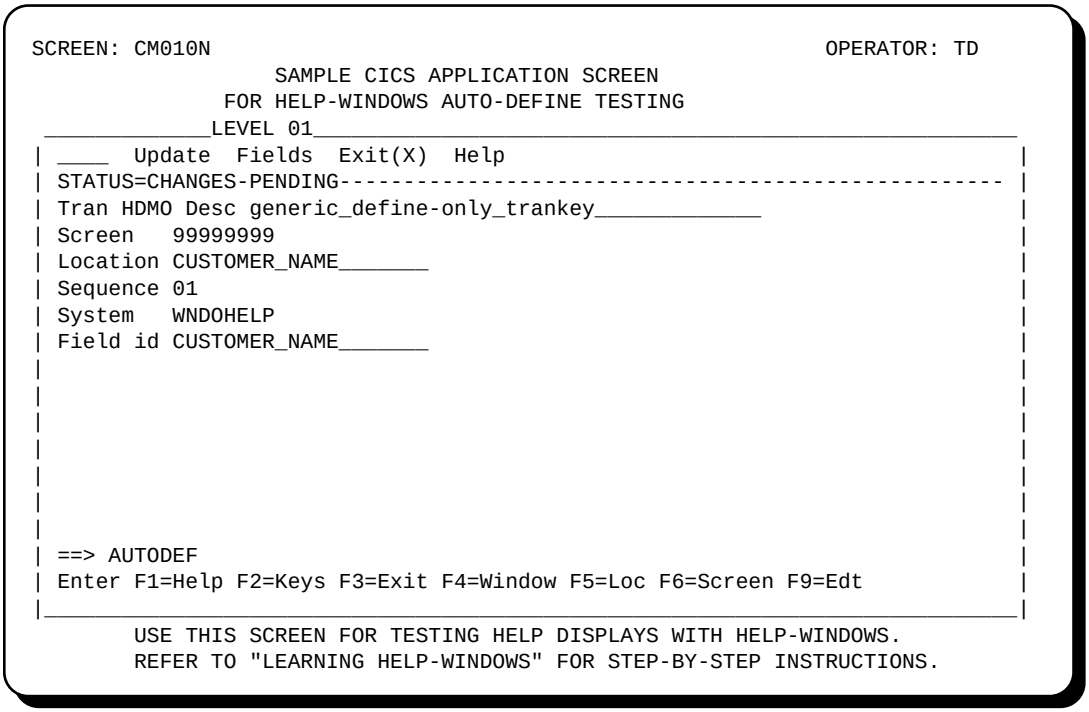
However, for this example, we will correlate a literal on the screen to the help text that is to display for this field. For this method, position the cursor to the "C" in CUSTOMER NAME.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 3. PRESS THE HELP-DEFINITION HOT KEY

Now, without moving the cursor, press the PF or PA key that you designated as your "Define" key.

Upon pressing the help definition key, the following window will appear on the screen:



Explanation of results:

The size and position of the window is determined from the help definition that was last created for this transaction, if any. Otherwise it is determined from the distributed "Model" definition.

The STATUS information is provided to indicate the current status of this transaction key record. NEW RECORD indicates that a new help definition is being created and has not yet been added to the file.

The TRANSACTION is filled in with HDMO, the transaction in progress.

The SCREEN ID is all nines, indicating that a unique screen identification is currently not defined.

LOCATION contains CUSTOMER NAME. If we were defining absolute row/column cursor position help, this field would contain 07023, which is row 7 column 23.

The SEQUENCE field contains 01 since this is the first transaction key record to be defined at this position of this screen.

The SYSTEM field contains WNDOHELP, which is the default system ID. System names are used merely for grouping different help subjects together. This is explained in more detail under *THE AUTODEF WINDOW.*

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The FIELD ID contains CUSTOMER NAME. This field is used as a cross document name. Cross documents are explained later under *THE AUTODEF WINDOW*.

The ==> in the window is the window command area. It currently contains the AUTODEF command, which must be there in order to respond to the display using the PF key prompts on the bottom row.

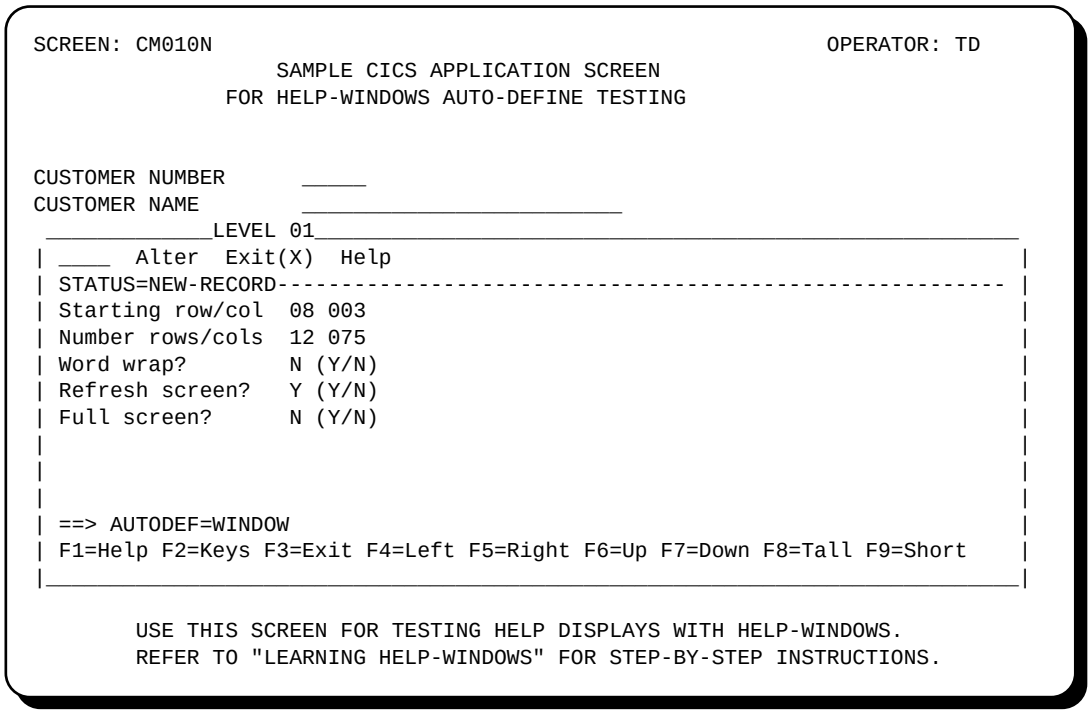
The bottom row of the window contains the directional options (key prompts) which are available at this stage of auto-define.

STEP 4. CHANGING THE WINDOW

Since we're going to create a field help definition, we need to make the window smaller and place it so that the CUSTOMER NAME field can be seen.

Press PF4 to enter change-window mode. Then, either use the PF keys, window commands or direct entry of row/column information to size the window like you want it.

By setting the STARTING ROW/COL to '12 003', NUMBER OF ROWS to '08' and NUMBER OF COLUMNS to '075' we will have a window which appears as follows:

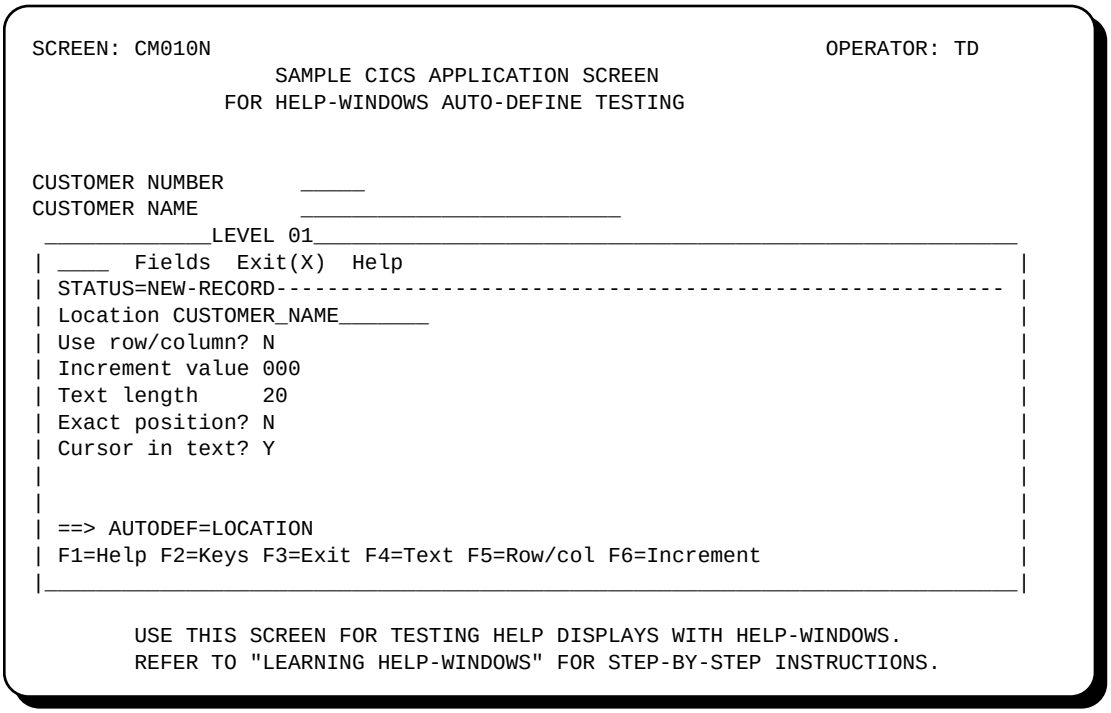


Now, press PF3 until the main AUTODEF display reappears.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 5. IDENTIFY THE HELP FIELD

Earlier we pressed the Define key while the cursor was positioned to the field descriptor. However, for help access, we will want the operator to be able to press the Help Access key while the cursor is in the actual data entry field. Now we need to specify the offset from CUSTOMER NAME to the actual data entry field. For this press PF5, the Location Window will appear as follows:



Now position the cursor to the first position of the actual data entry field and press PF6 (Increment), which will automatically compute the offset from the data entry field to the associated literal on the screen (the position of the cursor when the Define key was pressed).

Upon pressing PF6, the value '21-' should appear in the INCREMENT VALUE field. This indicates that the literal 'CUSTOMER NAME' appears 21 positions prior to the cursor location.

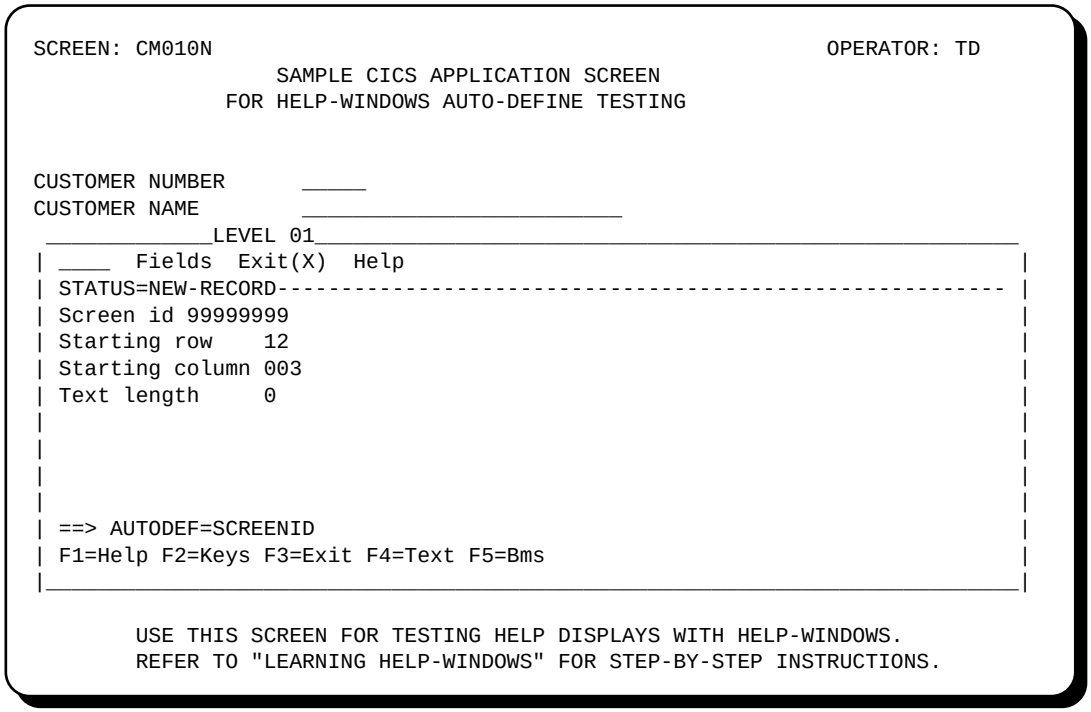
Note that if EXACT POSITION is left as 'N', the cursor can be anywhere in the customer name data field when the Help key is pressed, and still associate properly with the CUSTOMER NAME literal.

Now, press PF3 to redisplay the main AUTODEF window.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 6. IDENTIFY THE SCREEN

To ensure that this field level help is not accessible on a different screen of this transaction, to which this help text may not be exactly accurate, you can specify a screen identifier. To do this press PF6 from the main AUTODEF window. The Screen Identifier window will appear:



Now locate a literal on the screen that is specific to this screen at the position that it occurs. In this example we have an actual screen ID at row 1, column 10. Position the cursor to this literal and press PF4. This will paste the literal in the SCREEN ID field and insert the position of the literal in the appropriate fields.

Now, press PF3 to redisplay the main AUTODEF window.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 7. ENTER THE HELP TEXT

Now we need to key the text that is to display when the Help Access key is pressed. For this press PF9 (EDIT). A full screen text editor will display, with the following statements:

<STRTFLD CUSTOMER NAME>

<STOPFLD CUSTOMER NAME>

These statements define the beginning and end of the help text that is to display for this field. All help text must fall between these statements. The initial number of lines between these two statements is the number of lines that will display in the currently defined window size; however, the STOPFLD statement may be moved to accommodate more help text.

When we have keyed all of the text that we want, it might appear as follows:

```
____  List  Color  New  Delete  Exit(X)  Help
---+----10---+----20---+----30---+----40---+----50---+----60---+----70---+----
<STRTFLD CUSTOMER NAME>                                                    ***
The customer name field must contain the full name of the customer.        ***
If this is a company name, use the standard company name format for        ***
your entry.                                                                ***
                                                                           ***
If this is a person's name, enter the last name first, then a comma,       ***
then the first name, middle (if known) and any titles (Jr, Esq, ect.)      ***
                                                                           ***
                                                                           ***
<STOPFLD CUSTOMER NAME>                                                    ***
                                                                           ***
                                                                           ***
                                                                           ***
                                                                           ***
                                                                           ***
                                                                           ***
                                                                           ***
                                                                           ***
                                                                           ***
==> EDIT,SY=WNDOHELP,DO=FASTPATH,SE=001,REC=039000
WH14013. TEXT RECORD HAS BEEN CHANGED.
Enter F1=Hlp F2=Keys F3=Exit F4=Lst F5=Add F6=Cpy F7=Bwd F8=Fwd F9=Colr F11=Toc
```

Let's assume that this is all we want to say for this help display and it will fit in the window size displayed. Upon completion of keying the text, we press ENTER. This causes the transaction key record, with the text that we keyed, to be added to the file.

Now press PF3 to exit the text editor and return to the main AUTODEF display. We have now completed the creation of a field-level help display.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 8. EXIT AUTO-DEFINE MODE

To exit auto-define mode, press PF3. If changes have been made to the help specifications, a message will display asking if you wish to save the changes. Enter "Y" or "N" accordingly. Then the original HDMO display will reappear.

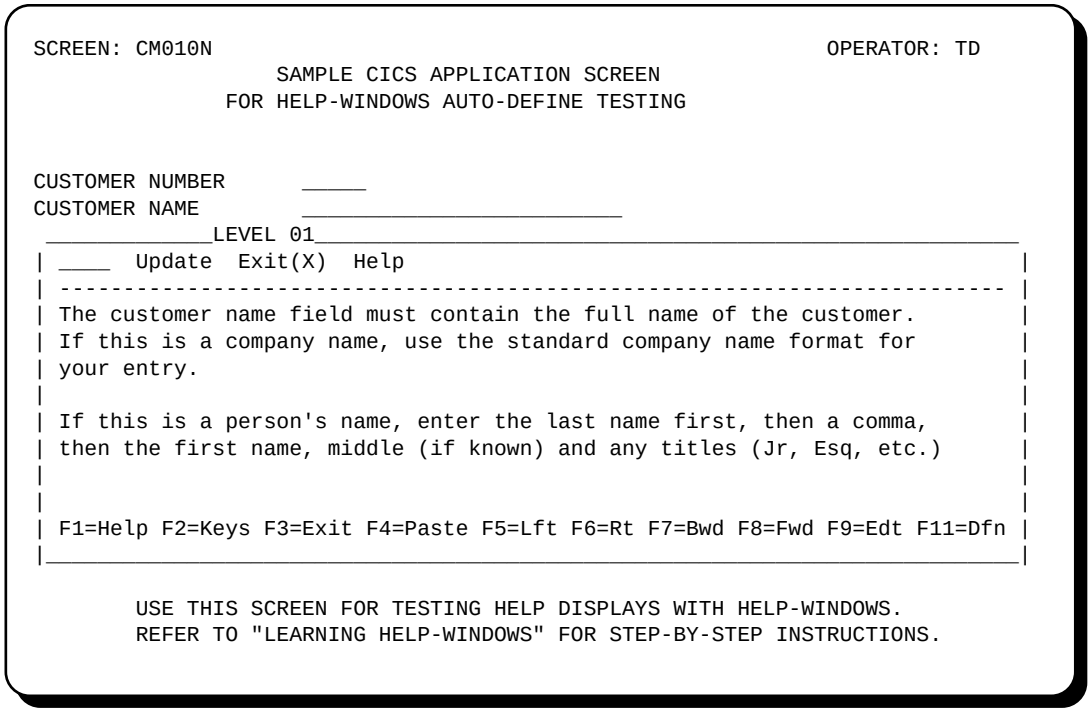
© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 9. TEST THE HELP DISPLAY

Now, position the cursor anywhere in the data entry field for CUSTOMER NAME, and press the Help Access key.

Upon pressing the help key, you should see the help display appear as the end-user will see it. That is, only the text displays in the window, with all control statements removed.

It should appear as follows:


SCREEN: CM010N OPERATOR: TD
SAMPLE CICS APPLICATION SCREEN
FOR HELP-WINDOWS AUTO-DEFINE TESTING
CUSTOMER NUMBER
CUSTOMER NAME
LEVEL 01
Update Exit(X) Help
The customer name field must contain the full name of the customer.
If this is a company name, use the standard company name format for
your entry.
If this is a person's name, enter the last name first, then a comma,
then the first name, middle (if known) and any titles (Jr, Esq, etc.)
F1=Help F2=Keys F3=Exit F4=Paste F5=Lft F6=Rt F7=Bwd F8=Fwd F9=Edt F11=Dfn
USE THIS SCREEN FOR TESTING HELP DISPLAYS WITH HELP-WINDOWS.
REFER TO "LEARNING HELP-WINDOWS" FOR STEP-BY-STEP INSTRUCTIONS.

CREATING A SCREEN LEVEL HELP DISPLAY

Screen level help is defined much the same as field level help; however, instead of Step 5 above, IDENTIFY THE HELP FIELD, you should simply code the LOCATION field of the main AUTODEF window with all nines (9) or erase it. This will specify a generic field ID, and will be accessed only if no field level help is defined for the cursor position at the time the Help key is pressed.

CREATING A TRANSACTION LEVEL HELP DISPLAY

Transaction level help is defined as screen level help, except that nines should be coded in the SCREEN field, as well as the LOCATION field of the main AUTODEF window. Transaction level help is only displayed if no field level help is available for the cursor position, and no screen level help is available for the current display.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CREATING HELP DISPLAYS, DETAILED REFERENCE

THE AUTODEF WINDOW

This is the first window to appear when the DEFINE key is pressed. Here is where you identify the transaction code and location on the screen where the cursor will be when the help key is pressed.

Default values are provided for all fields, depending on the location of the cursor when the DEFINE key was pressed, but any of these may be changed if desired.

Following is a description of each field:

STATUS The STATUS indicator shows the current condition of the definition record, whether it has been updated or not, etc.

Status messages can be ...

NEW-RECORD	-	A new help definition is being created.
CHANGES PENDING	-	Changes have been made and not yet recorded. Press Enter to update the definition.
NO CHANGES	-	No changes have been made to this definition.
RECORD CHANGED	-	The updates have been successfully performed.

In addition, various error messages can appear in the STATUS indicator. For example, when PF3 is pressed from the main AUTODEF window and changes are pending, a message appears asking if the changes are to be saved or not.

TRAN The TRAN field defines the CICS transaction code of the screen for this help display. It will be filled in from the transaction in progress when the DEFINE key is pressed, but may be changed.

Transaction codes may be generic. You can substitute a question mark (?) for any character in the trancode. The remaining characters that are not question marks must match exactly in the corresponding positions. If the trancode is a PF or PA key (TASKREQ option), code the mnemonic value for the key as PA1, PA2 or PA3, PF1, PF2 ... PF24.

[Note] The use of generic transaction codes should be carefully considered before using them. Additional overhead is incurred, for all non-generic trancodes are searched first. In addition, it is sometimes difficult to control and retrieve the correct help display due to confusion of similar transaction codes. New transactions in CICS may use trancodes that unintentionally match the generic help definitions.

DESC The Description field is not required but is recommended for ease in tracking and maintaining help definitions. You should briefly describe the field or screen of this help display.

The description will display on the help directory, which is obtained by entering transaction code WHLP or WHLP,D.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The description is an aid to find the help definition record for a given help display. It is particularly useful if you need to perform direct maintenance to the definition record (the transaction key record).

SCREEN

The SCREEN field is an additional means of qualifying a help display, besides the transaction code. You can create help displays that pertain only to certain screens of a transaction.

The field is filled with nines initially. This means that there is no unique screen identifier. To identify a screen, you must use either the BMS map name or some identifying text at a fixed location of the screen. Screen can be from one to eight positions long.

[Note] You should not use a unique screen identifier when defining nested help Nested help is help on help ... selections of a help menu, for instance, or further explanations of a given field. Nested help definitions should have all nines in the SCREEN field.

LOCATION

The LOCATION designates whether this is to be transaction-level, screen-level or field-level help. If LOCATION is filled with nines, it is either transaction-level or screen-level, depending on the contents of SCREEN.

Otherwise, this help display is to be invoked when the help key is pressed while the cursor is in or near a particular field of the screen.

The value in LOCATION is either a row/column display address, or some identifying text on the screen. For text on the screen, the cursor can be designated as IN the text, or in position RELATIVE to the text (either before or after it). Row/col is entered as a 5-digit number (RRCCC).

Text identification is recommended, wherever possible, because it offers much more flexibility for maintenance purposes. Also, text identification must be used for nested help definitions.

SEQUENCE

The SEQUENCE number is used for help window chaining. This allows the size and/or position of the help window to change automatically as the operator browses forward through the help text.

The initial sequence number of a new definition is always '01'. At any point, by increasing the value by one, a new definition record is created for the same transaction, screen and field identifiers.

By changing the window configuration for the new definition, subsequent help text will display in the new window. You can designate whether the old window is to remain on the screen (cascade effect) or disappear when the new window pops up by use of the Refresh Screen option of the Window alteration function (PF4).

If help window chaining is not desired, leave the sequence number as '01'.

SYSTEM

The SYSTEM field is the high-level qualifier of the help text. Help text is identified by five hierarchical qualifiers in the full HELP-WINDOWS product: SYSTEM, DOCUMENT, SECTION, SUBJECT, and RECORD NUMBER.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In the CICS-WINDOWS Help-Windows feature, the DOCUMENT, SECTION and SUBJECT are implicit. Document will always be FASTPATH, Section is '001' and Subject is blank. You can, however, define new SYSTEMS, which is a way to segregate your help text into logical sets.

If you're defining help for all transactions of the accounting system, for instance, you could define a new system called ACCTNG, enter all your help text, then define a different system for inventory. etc.

The default SYSTEM is WNDOHELP, which is supplied. To define a new system, simply enter a different system name.

FIELD ID — The FIELD ID is a unique name to be given to this set of help text. It is twenty positions long and may contain any characters, including imbedded spaces.

When the DEFINE key is pressed, the text at the cursor location is the default value placed in FIELD ID, if there is any. If this is not the name you want to assign to this field text, enter a different name.

When the EDIT key (PF9) is pressed, to enter the help text, the FIELD ID is used to automatically construct the STRTFLD and STOPFLD statements, which are used to bracket the help text for this display. The field is not considered to be 'defined' until a text record is created containing a STRTFLD and STOPFLD statement using the FIELD ID. At that point, a pointer record called the Cross Document (XDOC) record is created.

The names used for FIELD ID also display in the Table of Contents, when the table of contents is displayed in Define mode.

DIRECTIONAL OPTIONS AT THE AUTODEF WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 — (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 — (EXIT) Use this key to exit from the current display to the previous logical level.

PF4 — (WINDOW) Pressing this key will display the Window Sizing window. This display is used to specify the window size and position, whether the help display should appear as a full screen (rather than a window), etc. For more information, see ALTERING THE WINDOW SIZE AND POSITION, later in this section.

PF5 — (LOC) Pressing this key will display the Field Location window. This display is used to specify the location of the field on the application screen for which this help display is to be associated. For more information, see IDENTIFYING THE FIELD LOCATION, later in this section.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PF6 (SCREEN) Pressing this key will display the Screen Identifier window. This display is used to ensure that this help display can only be accessed from a screen (or screens) with the same identifying information. For more information, see IDENTIFYING THE SCREEN, later in this section.

PF9 (EDIT) Pressing this key will display the full screen text editor, which is used to create the help text that is to display. For more information, see THE HELP TEXT EDITOR, later in this section.

ALTERING THE WINDOW SIZE AND POSITION

This window appears when PF4 is pressed from the AUTODEF window. It's purpose is to change the size and/or position of the help window on the screen. In addition, it defines how text will display in the window, if previous windows are to be removed when this one displays, and whether the text should display in full-screen or window mode.

The current values in the fields, which describe the current configuration of the window, are derived from the model window that was located when the DEFINE key was pressed. This can be the original distributed model, or it will use the values from previous help defined for this transaction.

METHODS OF ALTERING THE WINDOW

You can change the window configuration in any of three ways:

1) Use PF keys PF4 through PF11 to adjust the window left, right, up, down, taller, shorter, wider or narrower. These keys move one row or column at a time until a window adjustment command is issued.

2) Issue a window adjustment command in the command area (==>). These commands are WL, WR, WU, WD, WT, WS, WW and WN, for left, right, up, down, taller, shorter, wider or narrower. You can optionally follow the command with =xx where xx is the number of rows or columns to move.

3) Directly change the Starting row/col and/or Number rows/cols fields to position the window where you want it.

STARTING ROW/COL

The Starting row/col defines the row/column address of the upper left-hand corner of the window.

The row field is two positions and may contain a value from one to 43.

The column field is three positions and may contain a value from one to 132.

The starting row/col will not automatically adjust for different screen sizes. If you are creating help for terminals where some are model 2 (24 x 80) and others are model 3, 4 or 5, you should tailor your help window position to the model 2 screens. Otherwise it will not display correctly.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

NUMBER ROWS/COLS

The Number rows/cols defines the depth and width of the window.

The first field is the number of rows that the window is deep. It is the number of text rows, including the action bar and the prompt line. Thus, if number of rows is set to 21, for instance, the maximum lines of help text that will display in the window is 18, since the action bar and prompt line require a total of three lines.

The second field is the number of columns the window is wide. This is a count of the number of text columns, not including the side window borders. The side window borders require five positions, so the maximum window width on a 80-character line is 75.

Number of rows can be up to 43, number of columns can be up to 132.

WORD WRAP The Word wrap field can contain a yes or no (Y or N) to specify whether word-wrap is to be activated for this window or not.

Word-wrap removes the requirement that the text in the window extend no wider than the window by causing the text to break on word boundaries when displayed in a help window. In other words, it allows you to create text wider than the window, but the operator will not have to pan left and right in order to read it all. They will only need to scroll down.

You can mix text that wraps with text that does not wrap, which is often desirable when columnar data is presented in a window. To do this, code an 'N' (no-wrap) in the line command field of the text editor for each line that is not to be wrapped.

REFRESH SCREEN

The Refresh screen field can contain a yes or no (Y or N) to specify if other help windows present on the screen are to be removed when this windows is displayed.

If 'Y' is coded, the original transaction screen is refreshed prior to displaying this window, thereby removing all lingering windows from other displays.

If 'N' is coded, this window will overlay the screen with no refresh, so that previously displayed windows will still appear. This allows you to create cascade windows that partially overlay one another.

The Refresh screen option is only applicable when you are defining nested help, such as a help menu, or further expansion of a current display.

FULL SCREEN

The Full screen field can contain a yes or no (Y or N) to specify whether the help text is to display in a window or not.

If 'Y' is coded, the text will display in full screen mode, with no borders or action bars.

If 'N' is coded, the text displays in a window, according to the size and position specifications of Starting row/col and Number rows/cols.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DIRECTIONAL OPTIONS AT THE WINDOW SIZING WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 (EXIT) Use this key to exit from the current display to the previous logical level.

PF4 (LEFT) Pressing PF4 will move the window left by one position. To move more than one position, enter WL=nn in the command area, where nn is the number of positions move.

PF5 (RIGHT) Pressing PF5 will move the window right by one position. To move more than one position, enter WR=nn in the command area, where nn is the number of positions move.

PF6 (UP) Pressing PF6 will move the window up by one position. To move more than one position, enter WU=nn in the command area, where nn is the number of positions to move.

PF7 (DOWN) Pressing PF7 will move the window down by one position. To move more than one position, enter WD=nn in the command area, where nn is the number of positions to move.

PF8 (TALL) Pressing PF8 will enlarge the window by one row. To increase by more than one position, enter WT=nn in the command area, where nn is the number of rows.

PF9 (SHORT) Pressing PF8 will shorten the window by one row. To decrease by more than one position, enter WS=nn in the command area, where nn is the number of rows.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

IDENTIFYING THE FIELD LOCATION

This window appears when PF5 is pressed from the AUTODEF window. Its purpose is to further define the screen location for which this help is to apply, or to define a new screen location for another help definition.

This function must be used if you are defining help with relative cursor position or if you want to change from text recognition to row/column or vice-versa.

If you are defining help using cursor in text, you can avoid this function by simply placing the cursor in the field to be documented prior to pressing the DEFINE key.

Fields can be associated with help displays using three methods. All pertain to the position of the cursor when the help key is pressed. The three techniques are:

Row/Column address	Absolute position on the screen
Cursor in Text	Cursor is in a field containing unique text.
Relative Cursor Position	Cursor is either preceding or following a field containing unique text.

Row/Column Address

To specify that this help is to be retrieved when the cursor is at a fixed location on the screen, you must use row/column recognition. At the Identify Location function, simply place the cursor at the desired spot and press PF5 (Row/col).

Alternatively, you can key the row/col address directly in the Location field of the AUTODEF window. It is coded as RRCCC, where RR is the row number and CCC is the column number (5 positions required).

Row/column recognition will locate the correct text if the cursor is anywhere within the field at the designated screen address. If you only want this help to appear if the cursor is at this exact location, you must code EXACT POSITION as 'Y'.

Cursor in Text

Text recognition provides greater flexibility than row/column, since it does not matter where on the screen the field appears. There must simply be some unique identifying text. The text can be up to 20 positions long and is coded in the Location field of the AUTODEF window.

To specify text recognition, place the cursor in the desired text and press PF4 (Text) at the Identify Location function. It does not matter whether the text on the screen is lower case or upper case.

Cursor in text recognition means that the cursor must be placed somewhere within the field containing the text. If Exact position is coded 'Y', the cursor must be positioned on the first character of the text for the help to be retrieved. To use cursor in text recognition, code a 'Y' in the CURSOR IN TEXT? field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Relative Cursor Position

Relative cursor position means that the cursor is placed somewhere before or after the identifying text. This is a common situation with many screens, when the field title precedes the data fields, such as:

NAME: _____

To specify relative cursor position, you must first use text recognition (place the cursor on the text and press PF4). Then you must specify an Increment factor, as a positive or negative number. A positive increment means the data field precedes the identifying text, where as a negative number means the data field follows it, as in the example above.

You can compute the increment factor yourself and key it, or you can let HELP-WINDOWS compute it for you, as follows:

1) Before pressing the DEFINE key, position the cursor on the first position of the text. In our preceding example, place the cursor on the 'N' of NAME. Now press the define key. 'NAME' will appear in the Location field.

2) Now press PF5 (Loc) to invoke the Identify Location function.

3) Move the cursor back to the first position of the data field.

4) Press PF6 (Increment). The correct positive or negative increment value will be placed in Increment value.

   This help display will now appear when the cursor is anywhere in the data field. To limit it to only the first position, code EXACT POSITION as 'Y'.

Following is a description of each of the fields of the Identify Location function:

LOCATION

LOCATION is the identifying text or row/column address where the cursor will be when the help key is pressed.

If text recognition is used, there can be up to twenty characters of text. The only limitation is that the text must be contained in a single field of the screen. There can be no imbedded attributes in the text.

If row/column recognition is used, LOCATION must contain a five-digit number in the form RRCCC, where RR is the two-position row number and CCC is the three-position column number (precede with zeros if needed).

USE ROW/COLUMN?

USE ROW/COLUMN must contain 'Y' if the value in LOCATION is to be a row/column address. If it is text, this field should contain 'N'.

If USE ROW/COLUMN contains 'N', even though the value in LOCATION is coded as RRCCC, it will still be considered to be identifying text.

When PF5 is pressed in the Identify Location function, this field is automatically updated with a 'Y'.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INCREMENT VALUE

INCREMENT VALUE contains a non-zero integer if Relative Cursor Position is being used. A positive number indicates that the cursor is positioned prior to the identifying text on the screen. A negative value means that the cursor appears after the text.

Negative numbers can be entered with the minus sign (-) either before or after the number. It is displayed with the minus sign following.

If PF6 (set increment) is pressed in the Identify Location function, the increment value is computed from the current cursor position to the cursor position at the time the DEFINE key was pressed.

[Note] For Row/column recognition, this field does not apply.

TEXT LENGTH

This field designates the number of characters in the screen field where the cursor will be when the help key is pressed.

For row/column recognition, the field length determines how far from the absolute row/column address the cursor can be and still considered a "hit" for this help text. This only applies if EXACT POSITION is coded 'N'.

For text recognition, field length is the number of characters in the identifying text to be considered. It must be equal to or less than the actual text length, but not greater.

With row/column recognition, you can specify a field length much longer than the true distance between attributes. This allows the same help text to be found for several like fields on a horizontal row.

EXACT POSITION?

EXACT POSITION must contain 'N' (no) if you want the capability to have the cursor anywhere within a field and still retrieve this help text. That is, the cursor does not have to be in the exact row/column address or positioned on the first character of identifying text to still work.

If coded 'Y', it works as follows for each method:

Row/column recognition	Cursor must be in the absolute row/column.
Cursor in text	Cursor must be on the first text character.
Relative cursor position	Cursor must be exactly the number of characters removed from the text as specified by the increment value.

CURSOR IN TEXT?

The CURSOR IN TEXT field designates, for text recognition, whether the cursor must be positioned in the field containing the identifying text or if it can be outside the field, using Relative Cursor Position.

Code 'Y' if the cursor is to be positioned somewhere in the text, otherwise code 'N'. If 'N' is coded, there must be a non-zero integer in the INCREMENT field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DIRECTIONAL OPTIONS AT THE FIELD LOCATION WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 (EXIT) Use this key to exit from the current help display to the previous logical level. (This may be an earlier help display or the transaction is progress.)

PF4 (TEXT) When using Cursor In Text or Cursor Relative To Text methods of defining help, you may position the cursor to the text on the application screen that is to be used for identifying this help display and press this PF key. This will place the text in the LOCATION field and the length in the TEXT LENGTH field.

PF5 (ROW/COL) When using the Row/Column method of defining help, you may position the cursor to the row/column position on the application screen and press this PF key. This will place the row/column in the LOCATION field and place a 'Y' in the USE ROW/COLUMN field.

PF6 (INCREMENT) When using Cursor In Text or Cursor Relative To Text methods of defining help, you may position the cursor to the field on the application screen in which the cursor should be when the Help Access key is pressed and press PF6. This will place the proper increment in the INCREMENT VALUE field.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

IDENTIFYING THE SCREEN

This window appears when PF6 is pressed from the AUTODEF window. It's purpose is to define this help definition as screen level or screen specific. In other words, this help definition is only to be invoked for a particular screen of a transaction, even though the field location information might match correctly on other screens.

There are two methods of identifying a screen. They are:

1) Unique text at a fixed screen location
2) BMS map name

Text identification is assumed. For BMS map identification, press PF5.

SCREEN ID

The SCREEN ID is the identifying text or the BMS map name which will uniquely identify this screen display.

The information in the SCREEN ID field is taken from the SCREEN field of the AUTODEF window. If this is an existing (already created) definition, changing the SCREEN ID will result in the creation of a new definition. In other words, you cannot change the SCREEN ID once a definition has been added to the file. You must create a new one, then delete the old.

The SCREEN ID can be one to eight characters in length and may contain any alphanumeric characters. To change the value in the field, you can key new data or move the cursor to the desired text on the screen and press PF4 (Text).

STARTING ROW

STARTING ROW is the row number on the display screen where the text in the SCREEN ID field will be found. It can contain a number from 1 to 43.

Be aware that HELP-WINDOWS will not automatically adjust for different screen sizes as it pertains to locating the screen identifier. For instance, if the identifying text appears at row 3, column 5 on a model 2 (24 x 80) screen, but row 1, column 35 on a model 5 (27 x 132) screen, you must make two definitions for all help unique to that screen. If you have this condition, try to find identifying text on row 1.

If you place the cursor on the first position of identifying text and press PF4 (Text), the starting row number will be automatically supplied.

STARTING COLUMN

STARTING COLUMN is the column number on the display screen where the text in the SCREEN ID field will be found. It can contain 001 to 132.

Be aware that HELP-WINDOWS will not automatically adjust for different screen sizes as it pertains to locating the screen identifier. For instance, if the identifying text appears at row 3, column 5 on a model 2 (24 x 80) screen, but row 1, column 35 on a model 5 (27 x 132) screen, you must make two definitions for all help unique to that screen. If you have this condition, try to find identifying text on row 1.

If you place the cursor on the first position of identifying text and press PF4 (Text), the starting column will be automatically supplied.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TEXT LENGTH

This field is the number of positions of the identifying text in the SCREEN ID field to be considered. Normally it would be the length of the data in screen ID, but it can be less.

To uniquely identify a screen, HELP-WINDOWS goes to the row/column address specified by STARTING ROW and STARTING COLUMN, compares the data at that location to the data in SCREEN ID for the length specified in TEXT LENGTH.

START OF MAP NAME

This field only occurs on the BMS screen identifier window (PF5 from the Screen Identifier window).

START OF MAP NAME designates the first position of the BMS map name to be used to identify the screen.

Normally this would be 1, but this option allows you to control one map name generically. It might be the case that you have several maps where the name is the same in the last four positions. You could define screen specific help that would apply to all those maps by designating the starting position as four and the length as four.

LENGTH TO USE

This field only occurs on the BMS Screen Identifier window (PF5 from the Screen Identifier window).

LENGTH TO USE designates the length of the BMS map name to be used to identify the screen.

Normally this would be the full name length, but you can control one map name generically. It might be the case that you have several maps where the name is the same in the last four positions. You could define screen specific help that would apply to all those maps by designating the starting position as four and the number of bytes as four.

DIRECTIONAL OPTIONS AT THE SCREEN IDENTIFICATION WINDOW

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 (EXIT) Use this key to exit from the current help display to the previous logical level. (This may be an earlier help display or the transaction is progress.)

PF4 (TEXT) You may position the cursor to the text on the application screen that is to be used for identifying this screen and press this PF key. This will fill all fields in the window to the appropriate values.

PF5 (BMS) If using a BMS mapped screen, press this PF key to display the BMS Mapped Screen Identification window.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

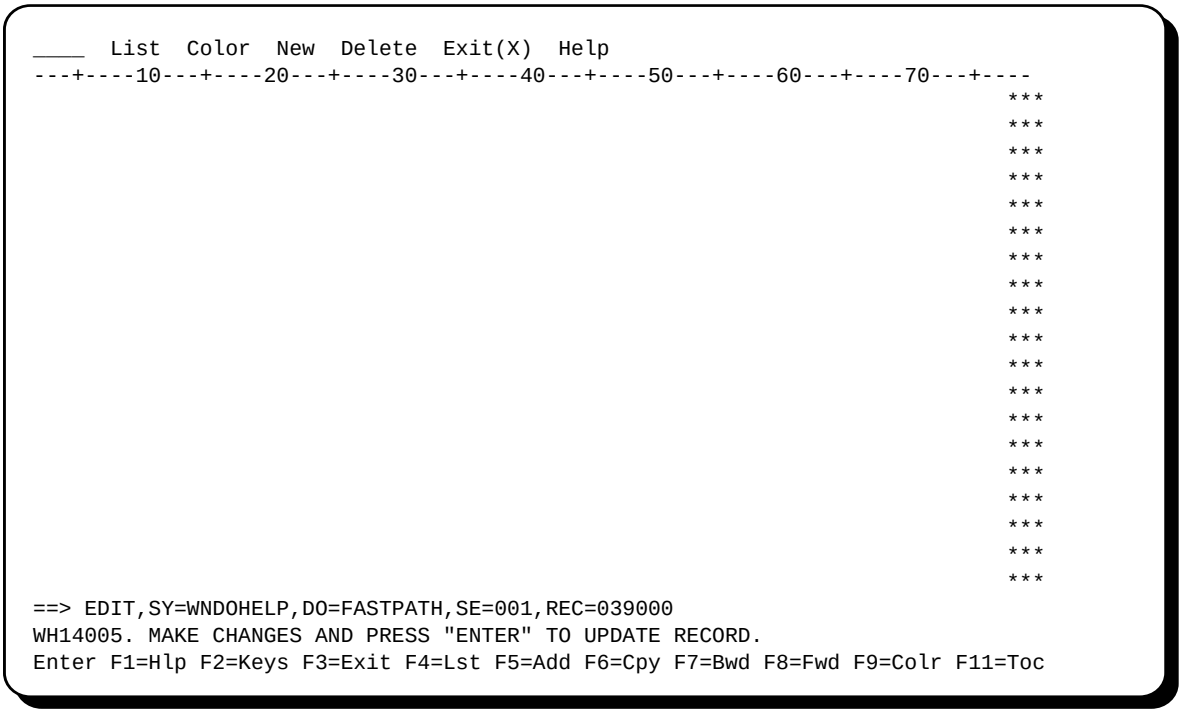
THE HELP TEXT EDITOR

The Help Text Editor is used for creating the display that the operator will see when the help key is pressed.

The help text is constructed in text records, each record containing 19 lines, one editor display page. When displayed in List form, or in a help display, the text is contiguous from record to record. Blank lines are always null, allowing full use of the INS key to shift data. Trailing blank lines at the end of a record are removed when the text is displayed in a help window.

When one text record is full, you must add another text record (PF5) in order to continue. Text records are automatically numbered, the record number displaying on the command line (==>) in the REC= keyword. You can move from one text record to another either by browsing forward or backward (PF8/PF7) or by changing the current record number.

This is used to 'paint the screen' the way you want the menu to be displayed. This screen appears when attempting to define or modify a menu by pressing the EDIT key from the Menu Directory or Menu Definition displays. This screen appears as follows:


____ List Color New Delete Exit(X) Help
---+----10---+----20---+----30---+----40---+----50---+----60---+----70---+----
***
==> EDIT,SY=WNDOHELP,DO=FASTPATH,SE=001,REC=039000
WH14005. MAKE CHANGES AND PRESS "ENTER" TO UPDATE RECORD.
Enter F1=Hlp F2=Keys F3=Exit F4=Lst F5=Add F6=Cpy F7=Bwd F8=Fwd F9=Colr F11=Toc

To enter text, simply key lines of data, formatting it as desired so far as line spacing, indentation, etc. are concerned. If changes are needed after keying some text, you can simply move the cursor to the area to be changed and type over any existing data. You can use the DEL key to delete characters in a line. To insert characters on a line, position the cursor where the insertion is to be made, press the INS key and type the desired characters. The ERASE EOF key may be used to erase all or part of a line at any time.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

When all text for this page has been keyed, press ENTER. This will write the updated text record to file. If the text being entered cannot fit on one screen, you may press the FORWARD key (or change the line number at the top of the display).

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

LINE DELETION

To delete an entire line, TAB to the COMMAND FIELD at the end of the line (where the asterisks are) and enter a 'D', then press ENTER. The entire line will be deleted and all following lines will shift up one line.

To delete more than one line at a time, enter a 'D' in more than one COMMAND FIELD before pressing ENTER or enter 'Dnn' on one or more COMMAND FIELDS, where nn is the number of consecutive lines (including the current line) to be deleted.

You may also perform block deletion by placing 'DD' on the first line to be deleted and another 'DD' on the last line to be deleted.

LINE INSERTION

To insert a line between two text lines, tab to the COMMAND FIELD of the preceding line and enter an 'I' over any one of the asterisks, then press ENTER. This will cause a blank line to be inserted following that line, and all subsequent lines will shift down one line. You may then key any desired text on that line.

To insert multiple lines, enter 'Inn' in the preceding COMMAND FIELD, where nn is the number of lines to insert, then press ENTER. This will cause nn blank lines to be inserted after the current line, and all subsequent lines will shift down nn lines.

MOVING AND COPYING TEXT LINES

One or more consecutive text lines can be moved or copied to another location. To move text lines, enter 'M' or 'Mnn' (nn = number of lines to move) in a COMMAND FIELD and press ENTER. This causes the specified line(s) to be deleted from that location. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER. The moved lines will be inserted following that line if the 'A' command is used, or before that line for the 'B' command.

To copy text lines, enter 'C' or 'Cnn' (nn = number of lines to copy) in a COMMAND FIELD and press ENTER. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER.

You may also perform block moves and copies by placing 'MM' or 'CC' on the first line to be moved or copied and another 'MM' or 'CC' on the last line to be moved/copied.

You may enter more than one 'M' or 'C' command on one edit screen, if desired. The result is that all selected lines are grouped together consecutively and inserted as one set wherever the 'A' or 'B' command is placed. For instance, if 'C2' is entered in the command area at line 3, then 'C4' is entered in the command area at line 16, all six text lines will be grouped together and inserted as one set following the line where the 'A' command is placed.

Note that you may use the 'A' or 'B' sub-commands as many times as desired, without performing another 'M' or 'C' sub-command. Each time it is used, the same data that was moved or copied last will be inserted at the new location.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TEXT RECORD SPLITS

When a text record is too full to contain more data, or any time you need to make room to insert additional text, you can perform a text record split.

To perform a split, place an S in the command field opposite the text line you want as the first line of the next record, then press ENTER.

A new text record will be added following this record. All text of the previous record from the line containing the 'S' to the end will be deleted from the previous record and added starting at line 1 of the new record.

INSERT STRTFLD/STOPFLD

The F command can be used to insert a STRTFLD or STOPFLD statement at any point in the text. To use it, do the following:

Place the 'F' in the command field of the line after which you want the STRTFLD or STOPFLD statement, and press ENTER.

If there is no STRTFLD in this text record, a formatted STRTFLD will be inserted with no field name. You must key the field name.

If a STRTFLD already exists prior to this point in the text record, a formatted STOPFLD will be inserted with the field name found in that STRTFLD. This avoids the necessity of keying the STOPFLD and ensures that the field name will be correct.

DISPLAY ONE OR MORE BLANK LINES - SPACE

The SPACE command is entered on a text line by itself (in position 1) in the form:

<SPACE nn>

where nn is the number of blank lines to display at this point. The nn value can be entered as a single digit.

SKIP THE DISPLAY TO THE NEXT PANEL - EJECT

The EJECT command is entered on a text line by itself (in position 1) in the form:

<EJECT>

The EJECT command forces a panel break at this point in the display. In other words, all text following the EJECT will display at line one of the next help panel when browsing forward in the help display.

EJECT also forces panel breaks when browsing backward. It is useful, therefore, for maintaining proper alignment of help panels during forward and backward browse operations.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DEFINE THE BEGINNING OF TEXT FOR ONE FIELD - STRTFLD

The STRTFLD command is entered on a text line by itself (in position 1) in the form:

<STRTFLD XXXXXXXXXXXXXXXXXXXX>

where xxx...xxx is the FIELD ID to be assigned to this block of text. The text is terminated with a STOPFLD statement containing the same Field Id. Field ID can be up to twenty characters long and contain any data.

The presence of a set of STRTFLD/STOPFLD statements with the same field ID causes the automatic creation of a special record called a Cross-document record. This record is keyed by the field ID and contains the starting and ending text record numbers for this text. The Cross-Document record is the means Help-Windows locates text.

You can nest sets of text by inserting STRTFLD/STOPFLD pairs around texts which is itself included in another STRTFLD/STOPFLD pair. There is no limit to the level of nesting. The only rule is that the Field ID on each STRTFLD/STOPFLD pair be unique.

Blocks of text delimited and named by STRTFLD/STOPFLD pairs can be the target of direct help displays by referencing the STRTFLD name in the Field ID of the AUTODEF window.

In addition, they can be the target of indirect displays by referencing the STRTFLD name on an INCLUDE statement.

DEFINE THE END OF TEXT FOR ONE FIELD - STOPFLD

The STOPFLD command is entered on a text line by itself (in position 1) in the form:

<STOPFLD xxxxxxxxxxxxxxxxxxx>

The STOPFLD statement marks the end of a set of text, the beginning of which was marked with a STRTFLD statement with the same name.

If the STOPFLD statement is omitted, the Cross Document record may be incomplete. The starting text record number will be recorded from the STRTFLD statement but the ending number is left zero. This means that only the first text record will display, then the Field ID will be considered to be at the end.

You can add missing STOPFLD statements at any time, plus move them from one text record to another using the M line command.

COPY AND DISPLAY THE TEXT FROM ANOTHER FIELD ID - INCLUDE

The INCLUDE command is entered on a text line by itself (in position 1) in the form:

<INCLUDE xxxxxxxx,yyyyyyyyyyyyyy>

where xxxxxxxx is the text SYSTEM name and yyy...yyy is the FIELD ID.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INCLUDE will retrieve the referenced block of text and insert it in-line when this point is reached as the text is being displayed. There can be any number of INCLUDE commands with or without normal text, and INCLUDE commands can be nested. That is, text retrieved with INCLUDE can have another INCLUDE within it. INCLUDE commands can be nested up to five levels.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

When using text attribute characters, set the attribute code where you want it to appear. Terminate it by repeating the same code where you want to return to normal intensity protected.

Unless otherwise specified, all words on the menu will be displayed in low intensity protected. In order to emphasize certain words or lines, you can use either of the following two methods:

1) Bracket selected text with text attribute characters. For this method, place one of the following characters before the text that is to be highlighted. The attribute will remain in effect for the remainder of the line or until another attribute is encountered. If another attribute of the same type is encountered, highlighting will revert back to the default (protected, low intensity).

   There are three text attribute characters, as follows:

Logical 'not' sign ()	Protected, high intensity
Pound sign (#) -	Unprotected, high intensity
'At' sign (@)	Unprotected, normal intensity

2) Use the Color pull-down menu or press PF9 while in the editor to invoke the attribute set function. This method is described on the following pages.

To invoke the Attribute function, press the ATTRIBUTES key (shown in the PF key prompt area). This places the terminal in SET COLOR mode. Note that upon invoking the attribute function, the function key area changes to include additional functions. These additional functions are as follows:

ENTER	If the cursor is inside the attribute window, pressing ENTER will move the cursor to the current attribute pointer position.
F3=EXIT	Exit Set Attribute mode to the editor.
F4=SET	Set an attribute (as specified in the attribute window) at the current cursor position.
F5=SELECT	Select the attribute at the position of the cursor and display the type of attribute in the attribute window. (If the cursor is inside the attribute window, this will move the window out of the way.)
F6=REMOVE	Remove (delete) the attribute at the position of the cursor.
F7=BWD	Move the current attribute pointer to the previous attribute.
F8=FWD	Move the current attribute pointer to the next attribute.
F9=EXTEND	Toggle the attribute window between CUA and extended attributes.
F11=SHOW	Show all attributes on the screen as an at-sign (@).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

When the attributes function is invoked, a pop-up window appears. The attributes window can appear in two forms:

```
CUA standard attributes
_ 1. Panel title
  2. Column heading
  3. Field prompt
  4. Data entry field
  5. Unprotected selection
  6. Protected selection
  7. Text
```

```
  Color           Protect
_ 1. Blue       _ 1. Protect
  2. Red          2. Unprotect
  3. Pink
  4. Green        Highlight
  5. Turquoise  _ 1. Normal
  6. Yellow       2. Blink
  7. White        3. Reverse
  8. Normal       4. Underscore
```

- A CUA standards attribute window. This window contains the CUA attributes for creating a display. The following chart portrays the display attributes for the various selections:

Selection	Extended color	Intensity	Protect/unprotect
Panel title	Yellow	High	Protected
Column Heading	Turquoise	High	Protected
Field Prompt	Turquoise	Normal	Protected
Data Entry	Green	Normal	Unprotected
Unprotected Selection	White	Normal	Unprotected
Protected Selection	White	Normal	Protected
Text	Blue	Normal	Protected

- An extended attribute window. This window offers all attribute combinations.

If the CUA standard window is displayed and you wish to use the extended attribute window, press the EXTEND key (shown in the PF key area). This key works as a toggle between the two types of attribute windows.

Upon pressing the ATTRIBUTES key, either a percent sign or logical not sign may display somewhere on the screen. This is the Current Attribute Pointer. If an attribute is present at that position of the screen, it will appear as a percent sign, else it will appear as the not sign. You may press the FORWARD or BACKWARD keys to position the pointer to the next or previous attributes on the screen, or you may position the cursor to a suspected location of an attribute and press the SELECT key to move the attribute pointer directly to that position.

To set an attribute, simply select the choices within the attribute window by entering the number of the choice. Then position the cursor to a space before the word (or line) that is to contain the attribute features. Then press the SET key (shown in the function key area). The attribute will remain in effect until another attribute or the end of the line is encountered.

To remove an attribute, position the cursor to the attribute to be removed and press the REMOVE key (shown in the PF key area).

[Note] Setting color with either attributes windows may cause extended attributes to be generated in the screen display when viewed. You must be using a terminal that supports extended attributes and the COLOR and/or HILIGHT feature(s) must be set in the CICS terminal control table (TCT) entry (or EXTENDEDDS with RDO) in order to view them. If you view a display on a terminal that does not support extended data streams, the extended color and highlighting attributes will be removed (in favor of regular field attributes) when the screen is displayed.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FINDING ATTRIBUTES ON THE SCREEN

Since attributes are normally invisible it is sometimes difficult to locate them in order to change or remove them. To do this, use the Show attribute function, which can be invoked either from the Show pull-down menu or by pressing PF11 when the attribute window is up.

Show attributes will show the location of all attributes on the screen by placing an at-sign at the attribute position. You can then move the cursor over an attribute and remove it with PF6 or reset a different attribute with PF4.

You can also use PF7 and PF8 to locate attributes either before (PF7) the current cursor location or after (PF8). When pressed, a percent sign will appear at the attribute location and the cursor will stop there.

MARKING AN ATTRIBUTE LOCATION - THE SELECT KEY

When the cursor is positioned where you want to place an attribute but you need to change the attribute to be set, you can save time by first pressing the Select key (PF5). This will display a not sign at the cursor location. Now tab to the attribute window and change the number of the attribute to be set.

When you subsequently press Enter, the cursor will jump back to the not sign. Now press PF4 to set the attribute.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key.

ENTER Update the menu or message and/or apply any commands in the command field (to the right of the screen).

PF1 HELP Display a Help screen for information pertaining to the text editor.

PF3 EXIT Exit to the previous logical level.

PF4 LST Display the text in List form. Text will display as it will appear to the end user, though still in full screen mode. All record breaks are removed, the text is contiguous, and all imbedded text commands (SPACE, EJECT, etc.) are honored.

PF5 ADD Add a new, empty text record immediately following the one currently displayed. The new record is then presented, ready to receive additional text.

The record sequence number for the new record is computed by reading ahead to the next record and halving the distance between its number and the number of the current record. If a new record cannot be inserted (sequence numbers differ by less than two), the text records must be reorganized in order to add another at this point. This can be accomplished by copying the next record to a new number, then deleting it, thereby making room, or by performing a Renumber command. See Miscellaneous Commands and Functions

PF6 COPY Copy the entire contents of the current text record and add it wherever you choose. After pressing PF6, you will be asked to press ENTER to complete the copy. At that point, you must change something on the command line, usually only the record sequence number, but you could also direct this text record to a different SYSTEM, if desired.

The COPY command accomplishes the same effect as doing a line copy for all lines of a text record, then doing an Add at the appropriate point in the receiving text, followed by a line insert of the copied text.

PF7 BACKWARD
Display the previous page of text.

PF8 FORWARD
Display the next page of text.

PF9 COLR Set the editor to attribute mode. This will display a different function key area at the bottom of the screen. For more information, please refer to CREATING COLOR AND HIGHLIGHTING earlier in this section.

PF11 TOC Display the Table of Contents.

While in the editor, the table of contents produces a different display than that seen while in help mode. The help mode table of contents is only those text lines that are marked as a TOC entry. The editor table of contents includes those, if any, but also displays a listing of cross-document field names. This will include all field names that have been defined via STRTFLD statements.

When the table of contents is displayed, you can tab to any line and press PF9 to edit the text associated with that field ID.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The edit table of contents makes it easy to locate text, or find the field ID that was used to define it.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

COMMANDS OF THE EDITOR COMMAND LINE

The command line of the editor is denoted by ==> at the bottom of the screen. All commands on the command line must be followed by the text record qualifiers, which are the keywords identifying the SYSTEM, DOCUMENT, SECTION and RECORD NUMBER.

These keywords are:

SY=system,DO=document,SE=section,REC=record number

The editor command itself must precede the first keyword. The following describes the commands available for managing the text editor. Note that all commands may be abbreviated down to two characters, and you can shift the command line left and right with the INS and DEL keys.

ADD	Add an empty text record at the designated number. This is the same as pressing PF5 while in the editor.
BACKWARD	Browse backward through the text file. Same as pressing PF7.
COLOR	Invoke the attribute set window for assigning color or highlight attributes to text. Same as pressing PF9.
COPY	Copy all text from this record and create a new record, adding it at a designated location. After entering the COPY command, you will be prompted to press Enter to add. At that point, the command line must specify the target, system, document, section and record number. Note that you cannot copy text records which contain a STRTFLD, as that would create a duplicate Field ID.
DELETE	Delete the current text record. After entering the delete command, you will be prompted to press ENTER to complete it. At that time, the record number on the command line is be deleted.
EDIT	Accept all input in edit mode, rewrite the text record with any changes and process all editor line commands. Also performs validity checks on imbedded text commands.
FORWARD	Browse forward through the text file. Same as pressing PF8.
HELP	Display a help screen for editor instructions. Same as PF1.
LIST	Display this and following text records in list mode, as it will be viewed by the end-user. Trailing blank lines are dropped and imbedded text commands are interpreted. Same as pressing PF4.
KEYS	Display the list of PF key assignments available while in the editor. This is the same as pressing PF2.
QUIT	Exit the editor and return to previous level. Same as PF3.
RENUM	Renumber all text records in this system. This command will assign new record numbers to all text records, incrementing each by 1000 or the value specified by the INCR=nnnn keyword, if present. For more information on renumbering text records, see MISCELLANEOUS COMMANDS AND FUNCTIONS, later in this section.
SEARCH	This command will continue a text search for a previously entered character string.
T	This command continues a search command in reverse order.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USING THE WHLP DIRECTORY

The directory is provided to quickly locate and optionally change or copy the various parts of a help definition.

There are three directories available. These are:

1) The transaction directory. A display of all TRANSACTION KEY records.
2) The field directory. A display of all CROSS-DOCUMENT records.
3) The system directory. This is a display of all text SYSTEM names.

The directories are invoked with the WHLP transaction code. If it is entered with no operands, the Transaction Directory will display, starting with the first transaction definition on file.

TRANSACTION DIRECTORY COMMANDS

WHLP	Start display with first trancode
WHLP,D	Start display with first trancode
WHLP,D,TR=xxxx	Start with trancode xxxx
WHLP,D,TR=xxxx,SCR=yyyyyyyy	Start at tran xxxx, Screen yyyyyyyy
WHLP,D,TR=xxxx,SCR=yyyyyyyy,LOC=zzz...	Tran x, Screen y, Location zzz..zzz

[Note] To position to a particular location, TR= and SCR= must be fully entered, but LOC= does not. The directory will start with the closest to whatever is entered on the LOC= keyword.

The transaction directory is a listing of Transaction Key records. These are automatically created when an Auto-Define function is performed.

Once the transaction directory is displayed, you can browse forward and backward with PF8/PF7. After tabbing down to an entry, you can perform any of the following functions on that entry by pressing a key

Enter	-	Bring that entry to the top of the display.
PF2	-	Copy that transaction key record using new key information.
PF4	-	Display the transaction key record for update.
PF6	-	Delete that transaction key record
PF9	-	Edit the text associated with that transaction key record.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELD DIRECTORY COMMANDS

WHLP,D,FI	Start display with first field definition
WHLP,D,SY=xxxxxxxx,FI	Start with first field of system xxxxxxxx
WHLP,D,SY=xxxxxxxx,FI=yyy...yyy	Start at system xxxxxxxx, field yyy...yyy

[Note] To position to a particular field, System must be fully entered, but Field can be generic. The directory will start with the field closest to whatever is entered on the FI= keyword.

The Field directory is a listing of Cross Document records. These records are automatically created from the STRTFLD/STOPFLD statements in the text.

Once the field is displayed, you can browse forward and backward with PF8/PF7. After tabbing down to an entry, you can perform any of the following functions on that entry by pressing a key.

Enter	-	Bring that entry to the top of the display.
PF2	-	Copy that cross-document record using new key information.
PF4	-	Display the cross-document record for update.
PF6	-	Delete that cross-document record
PF9	-	Edit the text associated with that cross-document record.

SYSTEM DIRECTORY COMMANDS

WHLP,D,SY	Start display with first system record
WHLP,D,SY=xxxxxxxx	Start display with system xxxxxxxx

[Note] To position to a particular system, the SY= value can be generic. The directory will start with the system closest to whatever is entered on the SY= keyword.

The System directory is a listing of SYSTEM records. These records are automatically created when a new text system is entered in auto-define.

Once the directory is displayed, you can browse forward and backward with PF8/PF7. After tabbing down to an entry, you can perform any of the following functions on that entry by pressing a key

Enter	-	Bring that entry to the top of the display.
PF9	-	Edit the first text associated with that system record.

[Note] The remaining functions on the prompt line (Copy, Update, Add Delete) cannot be performed on system records. You can not apply direct maintenance to system records.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MISCELLANEOUS COMMANDS AND FUNCTIONS

Additional functions and features of the Help-Windows feature of CICS-WINDOWS appear below.

TRANSACTION KEY RECORD DIRECT MAINTENANCE

The transaction key record is the "bridge" between the screen display and the correct help text. It is defined automatically as you move through the various windows of the Auto-define process, and therefore never needs direct maintenance.

You can, however, fully define help definitions or modify existing definitions by locating the correct transaction key record and modifying it, or adding an all new transaction key record.

To retrieve a transaction key record for direct maintenance, display the transaction directory by entering WHLP,TR=xxxx, where xxxx is the transaction code desired. Then locate the correct record, either by the description field or the SCRN ID / FIELD NAME information. Position the cursor by that entry and press PF4 (update). The transaction key record will appear generally as follows:

```
____  Text  New  Delete  Exit(X)  Help
--------------------------------------------------------------------------------
               HELP-WINDOWS Transaction Key Maintenance
Transaction Id  WAUX
Screen Id       APPLICAT               Screen Name Locator Information:
Field at cursor 04010_______________      Use BMS map name?           N  (Y/N)
Sequence number 01                           Start of map name        0
Define only ?   N  (Y/N)                     Number of bytes to use   0
Description:                              or...
  WAUX_Application_Startup_field_Help___     Use text on screen?      Y  (Y/N)
Transaction level help key                   Row where text begins    03
Associated cross document key:               Starting column          035
  System Id     WINDOWS_                     Number of bytes to use   8
  X-Doc name    ____________________
Text Display Specifications:           Field Name Locator Information:
  Use full screen?          N  (Y/N)      Use cursor row/column?      Y  (Y/N)
  If no, then ...                         If no, then...
    Row/col start of window 05 / 036        Is cursor in field text?  N  (Y/N)
    Rows/columns in window  11 / 042        Increment/decrement value 000
    Word-wrap desired ?     N  (Y/N)        Exact cursor position     Y  (Y/N)
    Refresh Screen ?        Y  (Y/N)        Length of field on screen 08
==> UPDATE,TR=WAUX,SCR=APPLICAT,LOC=04010              ,SEQ=01

Enter F1=Help F2=Cpy F3=Exit F4=Lst F5=Add F6=Del F7=Bwd F8=Fwd F9=Edt F11=Wndo
```

The fields of the transaction key record will be familiar to you once you have performed Auto-define. Each of the set of fields from the various AUTODEF windows appear in four quadrants of the transaction key record.

All of the same rules apply to the fields of the transaction key record as for the Auto-define process. For help on individual fields, position the cursor to the desired field and press PF1.

The functions available by PF keys and pull-down menus are as follows:

PF2 (Cpy)	-	Copy this transaction key record with new key information.
PF4 (Lst)	-	Display the associated help text in list mode.
PF5 (Add)	-	Add a new transaction key record.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PF6 (Del) - Delete this transaction key record.

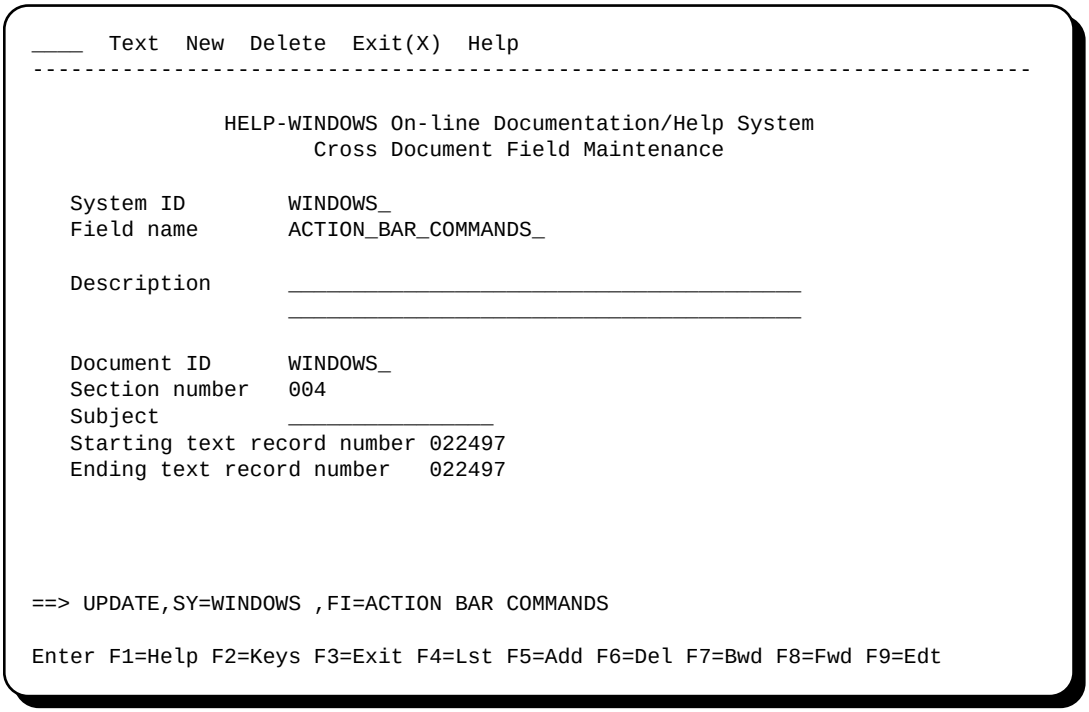
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PF7 (Bwd)	-	Browse backward
PF8 (Fwd)	-	Browse forward
PF9 (Edt)	-	Display the associated help text in edit mode.
PF11(Wndo)	-	Display the window as it will appear.

CROSS DOCUMENT RECORD DIRECT MAINTENANCE

The Field name is the same as the twenty-byte Field ID from the AUTODEF window and the name on the STRTFLD/STOPFLD statements in the text.

To retrieve a cross-document record for direct maintenance, display the Field directory by entering WHLP,FI or WHLP,SY=xxxxxxxx,FI=yyy...yyy, where xxxxxxxx is the SYSTEM name and yyy...yyy is from one to twenty characters of the Field ID. Locate the desired record and position the cursor by that entry, then press PF4 (update). The cross-document record will display.


____ Text New Delete Exit(X) Help
HELP-WINDOWS On-line Documentation/Help System
Cross Document Field Maintenance
System ID WINDOWS_
Field name ACTION_BAR_COMMANDS_
Description
Document ID WINDOWS_
Section number 004
Subject
Starting text record number 022497
Ending text record number 022497
==> UPDATE,SY=WINDOWS ,FI=ACTION BAR COMMANDS
Enter F1=Help F2=Keys F3=Exit F4=Lst F5=Add F6=Del F7=Bwd F8=Fwd F9=Edt

A Cross-document record is created automatically when a STRTFLD statement is entered in the text, using the identifier on the STRTFLD as the Field name. You cannot change this value.

The DESCRIPTION field is maintained for compatibility with the full HELP-WINDOWS product. You can enter a description here if you like, which will display on the Field directory.

DOCUMENT ID is maintained for compatibility with the full HELP-WINDOWS product. For cross-document records created by auto-define, the document ID must always be FASTPATH.

SECTION NUMBER is maintained for compatibility with the full HELP-WINDOWS product. For cross-document records created by auto-define, the section number must always be 001.

SUBJECT is maintained for compatibility with the full HELP-WINDOWS product. For cross-document records created by auto-define, the subject number must always be blank.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STARTING TEXT NUMBER is the text record number of the first text record containing the STRTFLD that has the same name as Field Name in the cross-document record. You should not make changes to this field. The correct way to change the starting text number is to move the STRTFLD statement to another text record. By manipulating the STRTFLD, Help-Windows will always keep the cross-document records in sync. If for any reason the starting text number is incorrect, it can be directly changed here.

The ENDING TEXT NUMBER is the text record number of the last text record containing the STOPFLD that has the same name as Field Name in the cross-document record. You should not make changes to this field. The correct way to change the ending text number is to move the STOPFLD statement to another text record. By manipulating the STOPFLD, Help-Windows will always keep the cross-document records in sync. If for any reason the ending text number is incorrect, it can be directly changed here.

The Cross-document record is an automatic record that is created when a STRTFLD statement is first entered in text. It is the record that records the beginning and ending text record numbers of the help text for a given Field ID.

Maintenance of the Cross-document record (sometimes call XDOC), is fully automatic so that it is never necessary to display them, but when there is a question, or if for any reason the starting and ending numbers are incorrect, the record can be directly updated.

The functions available by PF keys and pull-down menus are as follows:

PF4 (Lst)	-	Display the associated help text in list mode.
PF5 (Add)	-	Add a new transaction key record.
PF6 (Del)	-	Delete this transaction key record.
PF7 (Bwd)	-	Browse backward
PF8 (Fwd)	-	Browse forward
PF9 (Edt)	-	Display the associated help text in edit mode.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

RENUMBERING TEXT RECORDS

When enough text records have been added in one spot so that the increment between the last text record number and the one following is 'one', it is necessary to perform some reorganization to the text file, assuming more records need to be added at that spot.

There are two ways to make room for more text records:

1) Manually renumber one or more records.
   a) Add a new text record after the record which is blocking the way.
   b) Use Line Move commands to move all text from the previous record (the blocking record).
   c) Insert the moved text into the new record just added.
   d) The blocking record is now empty and can be used for new text.

2) Use the Renumber command.

   a) At any text record of the current SYSTEM, over type the EDIT command in the command line (==>) with the Renumber command. It can be abbreviated as RE, REN, RENU or RENUM. There must be a keyword SY=, DO= and SE= present with correct values.
   b) You will be prompted to press ENTER to perform a renumber, or exit with the Quit key (PF3).
   c) Upon pressing ENTER, the renumber operation will begin, which makes two passes of all the text in that system. The first pass counts the text records and scans the cross-document records for matching references. The second pass performs the reorganization of the text, which involves deleting and adding many records. A running pass counter and record counter appears at the bottom of the screen to report the progress of the renumber operation.

   [Note] Be sure Dynamic Transaction Backout is active for the WHLP transaction. If the system comes down during a renumber, the text file can be corrupted.

USING THE WORD WRAP FEATURE

Word-wrap provides the ability to display help text which is wider than the window, but avoid requiring that the operator pan right and left to read it all. Word-wrap causes the text to break on word boundaries as it approaches the right edge of the window.

This is often useful, but sometimes the end result is undesirable for the following reasons:

1) Text wraps 'up' as well as 'down'. Thus, text from one paragraph may merge onto the end of a previous paragraph when that is not desired.

2) Text indentations are lost, so text that is formatted in columnar displays, for instance, loses its formatting when it displays.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

These two problems can be overcome in most situations by using the word-wrap control commands in the editor. There are two commands which can be entered in the sub-command field (***) to the right of the text line to which they apply.

P (Paragraph end) Marks this text line as the last line of a paragraph so that subsequent text will not merge onto the line.

N (No wrap) Marks this text line as non-wrappable so that it will display exactly as it is keyed. This is often used when blank lines are left between paragraphs instead of a SPACE command. It also resolves the columnar data problem previously mentioned.

The word-wrap control commands will remain in the sub-command area unless they are removed by spacing over them. You may still enter other editor line commands (M, C, I, D, A, B) in the same field.

RESETTING UPPER CASE TRANSLATION

Upper case translation for the terminal is automatically set off any time the editor is entered so that help text may be keyed in lower case. It is restored when the WHLP transaction ends normally.

If an abend occurs, or a terminal error, or if for any reason the translation mode is not restored, a command is available which will reset it.

On a clear screen, hold the shift key down while you enter

WHLP,UC=ON

This will cause the upper case translation feature of the terminal to be restored so that normal operation can continue.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CREATING AND USING A TABLE OF CONTENTS

You can create a table of contents for an entire text system, which will display on-line from any help display. The table of contents is useful as a quick look-up for the operator to find the desired text, without constructing a lot of help menus. When the table of contents is displayed the operator can tab to the desired topic, and press ENTER to retrieve the help text for that subject.

Table of contents entries are keyed in the text as imbedded text commands, similar to STRTFLD/STOPFLD statements. The format is as follows:

<TOCx Text you want to display as a table of contents entry>

where 'x' is a 1, 2 or 3, designating the level of indention desired. TOC1 statements indent two positions, TOC2 statements indent by four, TOC3 will indent six positions. This allows you to specify topics and sub-topics.

The text of the TOCx command can be as long as the available space on the line, but remember to allow for window size and indentation. The table of contents will display in whatever window is in use when requested.

To display the table of contents in help mode, press PF11. The display will appear generally as follows:

```
This line would be a TOC1 table of contents entry
        This line would be a TOC2 table of contents entry
        This line would be a TOC2 table of contents entry
                This line could be a TOC3 table of contents entry
                This line could be a TOC3 table of contents entry
        This line would be a TOC2 table of contents entry

This line would be a TOC1 table of contents entry
```

Once the table of contents is displayed, the operator can press the TAB or RETURN key to position the cursor to the desired topic and press ENTER. The help text for that table of contents entry will display in the current window.

Table of Contents in Edit mode

In the editor, pressing PF11 will also display the table of contents, but in addition to any TOCx entries found, it will also display the Cross-Document Field names. These records are marked in the display as XDOC= to distinguish them from TOCx entries.

You can position the cursor to an XDOC entry and press ENTER to quickly move to the starting text for that field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SEARCHING FOR TEXT STRINGS IN THE EDITOR

It is often desirable to locate the position of one or all occurrences of a text character string, which may occur in any text record of the system.

This can be done with the text search command, which is appended to the end of the EDIT command on the command line as follows:

EDIT,SY=system,DO=document,SE=section,REC=nnnnnn S=xxxx...xxxxxx

where all characters following the S= keyword become the search string. Do not enclose the string in quotes. If imbedded spaces are present, they will be included in the search criteria. Upper and lower case characters are considered the same in text searches. The maximum length of the search string is the number of characters from S= to the end of the line.

If the search string is found, the text record where it is found is displayed with all text lines highlighted which contain the search string. The command now changes to:

SEARCH,SY=system,DO=document,SE=section,REC=nnnnnn

and a message is displayed indicating to press ENTER to continue. The search will continue until the end of the document is reached, stopping at each text record where the search string is found, as long as ENTER is pressed.

To stop the search on any text record, change the SEARCH command back to EDIT. You can then make any desired changes to the text record.

To search backward in the text file, enter the search string as T=xx..xx rather than S=xx..xx. The search will then proceed from the current record number in a reverse direction until the beginning of the document is found.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

IMPORTING HELP TEXT FROM ANOTHER SOURCE

If you have text in some other form that you wish to load into the text file for use by the Help-Windows feature of CICS-WINDOWS, that can be performed with the batch reformat program, WNDOREFM.

Use whatever means are available to get your text records into 80-byte card image form. For MVS, this can be a PDS member or a sequential file. For VSE it must be an unblocked SYSIPT file.

WNDOREFM supports a series of control statements that must appear at the beginning of your document. You can load several different documents in one pass, if desired, by placing the critical control statements at the appropriate point in the text record file.

The following describes the control statements and JCL for WNDOREFM.

Control statements input to WNDOREFM

&SYSTEM=xxxxxxxx

Denotes the following text as a new SYSTEM and establishes the system name. This may optionally be followed by a 79-byte system description, which must contain an ampersand in the first position.

&DOCUMENT=xxxxxxxx

Denotes the following text as a new DOCUMENT and establishes the document name. This may optionally be followed by a 79-byte description, which must contain an ampersand in the first position.

&SECTION=nnn

Denotes a new SECTION and sets the section number. May optionally be followed by a description record containing an ampersand in position one.

[Note] DOCUMENT and SECTION default to FASTPATH and 001 when text is added online. You may load text with different identifiers, however.

&CASE=UPPER|LOWER|MIXED

Sets the translation mode to upper case, lower case or mixed case. CASE=UPPER will force all text to upper. CASE=LOWER will force it to lower, CASE=MIXED leaves it untouched. Default is MIXED.

&REPLACE=YES|NO

Specifies if duplicate records already in the text file are to be deleted before loading this text. If NO, and a duplicate is found, a message is printed and the text record is skipped. REPLACE =YES should be used only to reload corrected data.

&SHIFTLEFT=FNB|nn|OFF

Shift all incoming text to the left to the first non-blank character or a specified number of positions. Can be used to eliminate text indentions.
&SHIFTLEFT=FNB justifies all text to position 1.
&SHIFTLEFT=nn shifts left nn positions.
&SHIFTLEFT=OFF performs no shifting.
The default is &SHIFTLEFT=FNB.

&START=nnnnn|1000

Sets the starting text record number. May be any free-form number from 1 to 99999. Default is 1000.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

&INCRE=nnnnn|1000

Sets the text record increment factor, the value to be added to each record number. Default is 1000.

[Notes]

1) All parameters stay in effect until replaced by another parameter with the exception of &REPLACE, which resets to NO when a new &SYSTEM, &DOCUMENT or &SECTION keyword is found.

2) All parameters except &SYSTEM, &DOCUMENT and &SECTION may appear at any point in the text and take effect at that point.

The 80-byte text records may contain imbedded text commands. These are

<STRTFLD xxxxxxxxxxxxxxxxxxx>	Define start of FIELD xxx...xxx
<STOPFLD xxxxxxxxxxxxxxxxxxx>	Define end of FIELD xxx...xxx
<SPACE nn>	Insert nn blank lines.
<EJECT>	Force help panel break.

JCL required for the WNDOREFM program

MVS example:

```
//TEXTLOAD  JOB   N,ACCOUNT-ID,MSGCLASS=X
//STEP1     EXEC  PGM=WNDOREFM
//STEPLIB   DD    DSN=XXX.XXX.XXX,DISP=SHR
//HWN$FIL   DD    DSN=XXX.XXX.XXX,DISP=SHR        CICS-WINDOWS    CONTROL
FILE
//SYSPRINT  DD    SYSOUT=*
//SYSUDUMP  DD    SYSOUT=*
//SYSIN DD  *
&SYSTEM=MYHELP
&This is a description of my system, which will appear in the directory.
&DOCUMENT=MYDOC
&SECTION=001
(80-BYTE CARD-IMAGE TEXT RECORDS)
/*
//
```

DOS/VSE example:

```
// JOB TEXTLOAD
// LIBDEF PHASE,SEARCH=LIBRARY.SUBLIBRARY
// DLBL HWN$FIL,'CICS-WINDOWS CONTROL FILE',,VSAM
// OPTION PARTDUMP
// EXEC WNDOREFM,SIZE=AUTO
&SYSTEM=MYHELP
&This is a description of my system, which will appear in the directory.
&DOCUMENT=MYDOC
&SECTION=001
(80-BYTE CARD-IMAGE TEXT RECORDS)
/*
/&
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

METHODS OF CREATING A HELP MENU

There are various ways to create menus of help displays. You can use any existing programming technique to build a menu or you can use HELP-WINDOWS to do it.

If you have an application development system or fourth-generation language package you could create a menu panel using the "paint-the-screen" technique available in most of these packages, then define a transaction key for a word which appears in each of the selection prompts of the menu screen.

You could accomplish the same thing with an application program using BMS mapping or any other screen-building technique.

For both of these techniques you should set an unprotected attribute code somewhere in the text of the selection prompt to allow the operator to tab the cursor to that position and press the help key. You then define a field-level transaction key for that position using either row/column specification or cursor-in-text specification.

You can also use HELP-WINDOWS to build a help menu through the use of nested help displays.

In general, you create a text record which will be your menu display, then define help definitions which are associated with a word in each of the selection prompts. You can use auto-define for these definitions but you will need to supply the transaction code of the dummy transaction (see USING DUMMY TRANSACTIONS, on the next page) by typing it at the main AUTODEF display. Each transaction key either references structured text or fast-path text to be associated with this menu selection. The text may be displayed in full-screen or window mode, at your discretion.

Now you must provide a method for the operator to get to this.

You could associate the help menu with some other user menu that the operator always sees, or you could associate it with a transaction screen. This would be accomplished through the normal means, that is, defining a help definition to be recognized on a user transaction or screen, with the text pointing to the help menu text record.

USING DUMMY TRANSACTIONS

Still another method is to use "dummy" transactions. This method provides a way to get directly to a help menu or directly to the text from a clear screen.

Since HELP-WINDOWS will recognize any text on the screen and can associate that with a help display, it is not necessary that the operator be in a real transaction in order to retrieve a help display. You can define a help definition and say that the transaction ID is anything, whether it is a real transaction or not.

For instance, you could define a transaction key with the transaction ID of HELP. HELP is not necessarily defined to CICS as a valid transaction code. From a clear screen, if you enter HELP, then press the help key (don't press ENTER), HELP-WINDOWS will retrieve and display the first page of text associated with that transaction. HELP-WINDOWS does not know or care whether it is a real transaction or not.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Your dummy transaction does not need to point to a menu, it could point directly to the help text. Thus, you could define a series of dummy transactions to invoke different on-line documentation displays. All the end-user needs to know in order to view the displays is what transaction to key.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

HELP ACCESS MODE - USING ON-LINE HELP DISPLAYS

This topic deals with the use of the HELP-WINDOWS feature by the end user. That is, the retrieval of help displays while in a user transaction. This is called Help Access Mode, or simply 'help mode'.

This section should be used by the text-writer for a deeper understanding of the kinds of specifications that can be made when defining help displays that will produce the various results discussed herein.

Command entry in help mode, screen chaining, multiple window displays, nested help and the cut/paste feature are discussed.

ACTIVATION OF HELP ACCESS MODE

Before a help display can be retrieved and displayed on the screen, help access mode must be "activated" for the terminal. Help access mode is activated when CICS-WINDOWS is activated at the terminal, if the CICS-WINDOWS user profile has a HELP KEY defined.

If CICS-WINDOWS is not on, or if the profile does not establish a Help key, the end user will not be able to retrieve help displays.

Activation of CICS-WINDOWS is performed with a WNDO,ON transaction, either entered directly at the terminal, or executed automatically using the automatic start-up feature of CICS-WINDOWS.

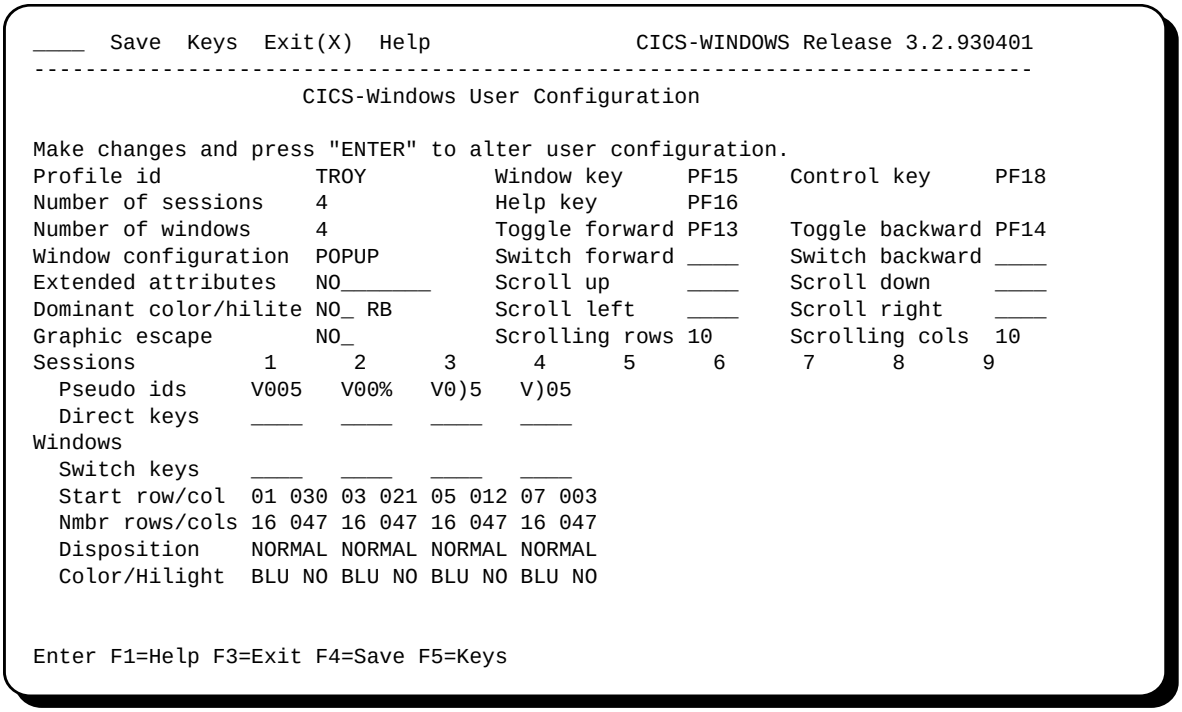
© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

RETRIEVING HELP DISPLAYS

To retrieve a help display, first initiate a user transaction for which a help display has been defined. Then proceed to the appropriate screen (if not already there) and press your help key.

We will use the WNIT transaction, as it is available to all CICS-WINDOWS users.

First we enter the WNIT transaction with no operands, resulting in the following display:


____ Save Keys Exit(X) Help CICS-WINDOWS Release 3.2.930401
CICS-Windows User Configuration
Make changes and press "ENTER" to alter user configuration.
Profile id TROY Window key PF15 Control key PF18
Number of sessions 4 Help key PF16
Number of windows 4 Toggle forward PF13 Toggle backward PF14
Window configuration POPUP Switch forward ____ Switch backward ____
Extended attributes NO_____ Scroll up ____ Scroll down ____
Dominant color/hilite NO_ RB Scroll left ____ Scroll right ____
Graphic escape NO_ Scrolling rows 10 Scrolling cols 10
Sessions 1 2 3 4 5 6 7 8 9
Pseudo ids V005 V00% V0)5 V)05
Direct keys ____ ____ ____ ____
Windows
Switch keys ____ ____ ____ ____
Start row/col 01 030 03 021 05 012 07 003
Nmbr rows/cols 16 047 16 047 16 047 16 047
Disposition NORMAL NORMAL NORMAL NORMAL
Color/Hilight BLU NO BLU NO BLU NO BLU NO
Enter F1=Help F3=Exit F4=Save F5=Keys

As distributed, this transaction has three types of help:

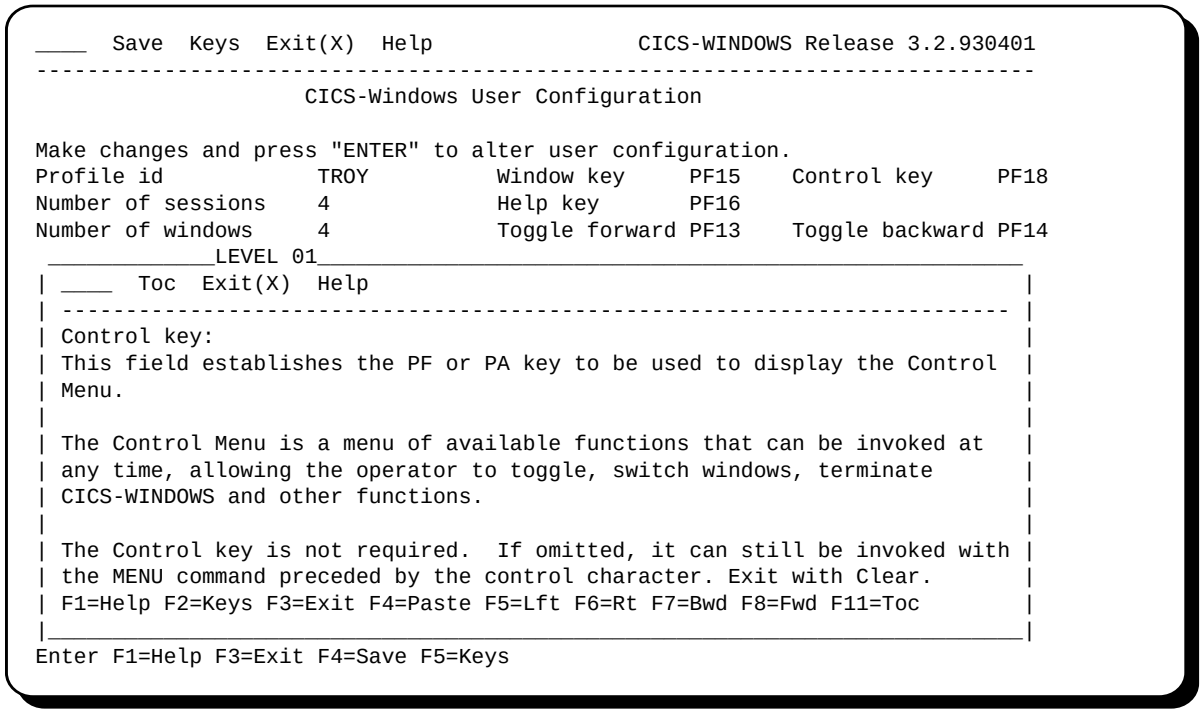
Field level help, which applies only to a certain field on the screen. This type of help is accessed by tabbing to the field in question and pressing the Help Access key.

Screen level help, which applies to the entire screen. This type of help is accessed by moving the cursor to an area on the screen for which no field level help is defined, (the title for instance). Screen level help may only be accessed if field level help is not available at the position of the cursor.

Transaction level help, which applies to the entire transaction. This type of help may only be accessed when the cursor is positioned to a field for which no help is defined on a screen for which no help is defined. This type is used for either of two reasons, for a fail-safe to ensure that help is always available, or for the only help display for very simple transactions.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Upon displaying this screen, tab to the Control Key entry field and press the Help Access key, the help text will display in a window as it was defined, as follows:


____ Save Keys Exit(X) Help CICS-WINDOWS Release 3.2.930401
CICS-Windows User Configuration
Make changes and press "ENTER" to alter user configuration.
Profile id TROY Window key PF15 Control key PF18
Number of sessions 4 Help key PF16
Number of windows 4 Toggle forward PF13 Toggle backward PF14
LEVEL 01
____ Toc Exit(X) Help
Control key:
This field establishes the PF or PA key to be used to display the Control Menu.
The Control Menu is a menu of available functions that can be invoked at any time, allowing the operator to toggle, switch windows, terminate CICS-WINDOWS and other functions.
The Control key is not required. If omitted, it can still be invoked with the MENU command preceded by the control character. Exit with Clear.
F1=Help F2=Keys F3=Exit F4=Paste F5=Lft F6=Rt F7=Bwd F8=Fwd F11=Toc
Enter F1=Help F3=Exit F4=Save F5=Keys

You will see the help window displayed for the Control Key field. An Action Bar appears at the top of each help window, providing pull-down menus for additional services. The last line of each help window is the directional options as set by PF keys. All information and commands are contained within the borders of the window.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DIRECTIONAL OPTIONS AT A HELP DISPLAY

The last line of the window is used to display the directional options available at this point. You can invoke one of these options by pressing the corresponding PF key. The available options are:

PF2 (KEYS) PF key prompts are displayed at the bottom of the window, however if the window is to narrow to display all available prompts, you may press PF2 to temporarily display the prompts in a vertical manner within the current window. Press Enter or PF3 to exit the Keys display, any other PF key will exit the keys display then perform the function shown in the Keys window.

PF3 (EXIT) Use this key to exit from the current help display to the previous logical level. (This may be an earlier help display or the transaction is progress.)

PF4 (CUT/PASTE) To use the cut/paste function, place the cursor anywhere in one of the words in the help display and press PF6. Upon pressing PF6, the help display will disappear and the application screen is re-displayed. The word where the cursor was placed is copied from the help display and placed into the field of the application display where the cursor was positioned when the help key was originally pressed.

PF5 (LEFT) Move the display to the left one or more positions. Pressing PF5 will pan left one position. To pan more than one position, enter LE=nn (or LEFT=nn) in the command area, where nn is the number of positions to pan.

Note that panning left means that you logically move the window to the left, over the text. It will appear as though the text moves to the right.

PF6 (RIGHT) Move the display to the right one or more positions. Pressing PF6 will pan right one position. To pan more than one position, enter RI=nn (or RIGHT=nn) in the command area, where nn is the number of positions to pan.

Note that panning right means that you logically move the window to the right, over the text. It will appear as though the text moves to the left.

[Note] If word-wrap is in effect for this display, there is no need to perform any panning. Word-wrap will cause sentences of text to break on word boundaries so that all of the text will display, no matter what size the window. With word-wrap, you only need to page forward to read all of the text.

PF7 (PREVIOUS) Display the previous page of text before this one. The display will move backward in the text for the depth of the window, displaying as much text as can be displayed in the window.

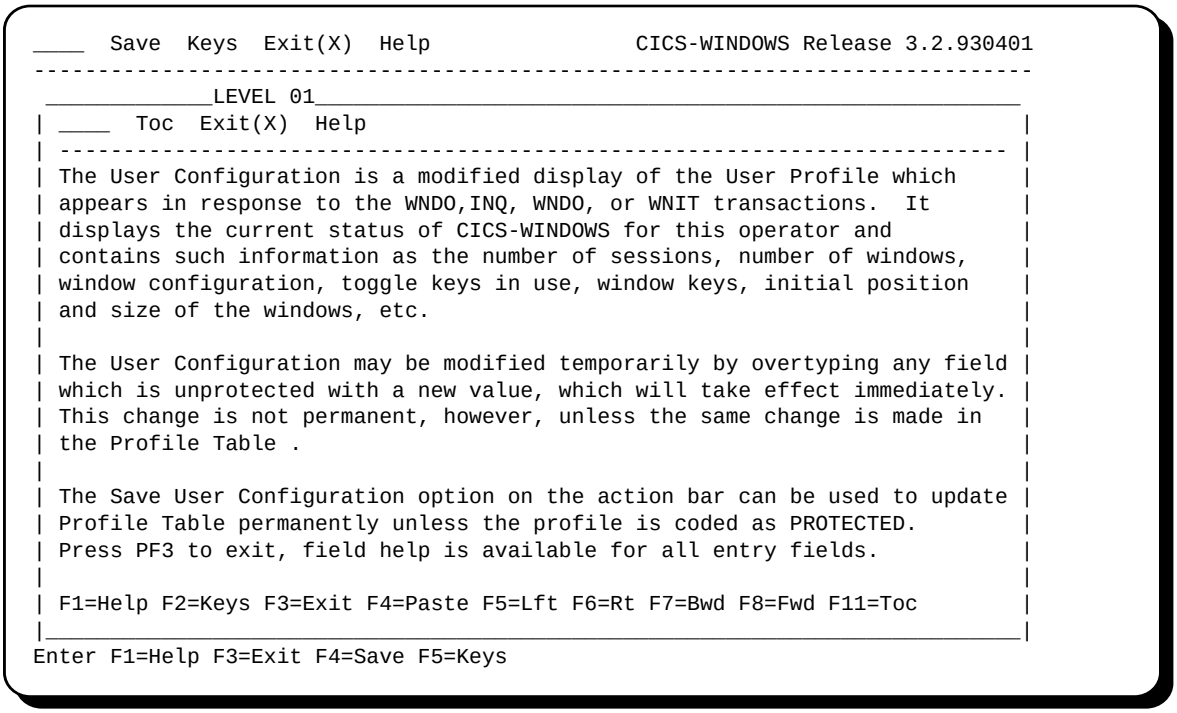
PF8 (NEXT) Display the next page of text after this one. The display will move forward in the text for the depth of the window, displaying as much text as can be displayed in the window.

PF11 (TOC) Display the Help Table of Contents.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

RETRIEVING A SCREEN-LEVEL HELP DISPLAY

The previous example illustrated the retrieval of a field-level help display. That particular display was retrieved because there was a field-level display defined. If the cursor was not positioned to a 'table' field at the time the Help Key was pressed (the screen title for instance), the following screen-level help would display:


____ Save Keys Exit(X) Help CICS-WINDOWS Release 3.2.930401
LEVEL 01
____ Toc Exit(X) Help
The User Configuration is a modified display of the User Profile which
appears in response to the WNDO,INQ, WNDO, or WNIT transactions. It
displays the current status of CICS-WINDOWS for this operator and
contains such information as the number of sessions, number of windows,
window configuration, toggle keys in use, window keys, initial position
and size of the windows, etc.
The User Configuration may be modified temporarily by overtyping any field
which is unprotected with a new value, which will take effect immediately.
This change is not permanent, however, unless the same change is made in
the Profile Table .
The Save User Configuration option on the action bar can be used to update
Profile Table permanently unless the profile is coded as PROTECTED.
Press PF3 to exit, field help is available for all entry fields.
F1=Help F2=Keys F3=Exit F4=Paste F5=Lft F6=Rt F7=Bwd F8=Fwd F11=Toc
Enter F1=Help F3=Exit F4=Save F5=Keys

When the help key is pressed, HELP-WINDOWS will determine that there is not a field-level help display defined for the current cursor position, however there is a screen-level display for a screen with the WNIT transaction code and the word "CONFIGUR" at row 3, column 40. The screen-level text will be retrieved and displayed.

The screen-level help is displayed in the window size and position that was defined. All of the same directional options previously explained are available at this point.

RETRIEVING A TRANSACTION-LEVEL HELP DISPLAY

As previously stated, transaction level help (if defined) will display only if no field or screen level help is defined. As distributed, the screens of the WNIT transaction have both field and screen level help, so transaction level help should never display. However, the directional options are the same for transaction level help.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CHAINED HELP WINDOWS

Chained help windows occur when multiple transaction key records are defined with the same transaction code, screen identifier and field identifier, with incrementing sequence numbers.

In this situation, each transaction key references a different section of text to be displayed, and may define a different window configuration.

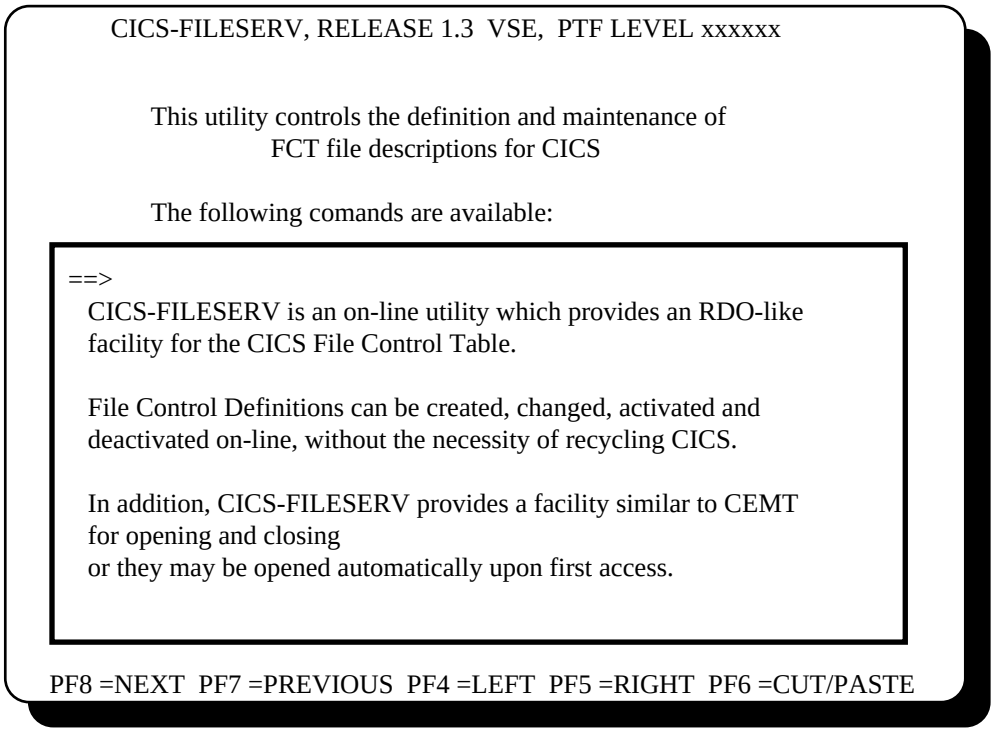
In help access mode, the text associated with the first transaction key will be retrieved when the help key is pressed. Upon browsing forward with the NEXT command or key, all of the text associated with the first transaction key is displayed. When the end of text is reached for the first definition, another NEXT command will cause the second transaction key record to be retrieved, whereupon the text and window configuration for that definition is displayed.

If the window configuration of the second transaction key is the same as the first, you cannot tell that chaining has occurred. If however, the second key has a different size and/or position for the window, you will see a different display.

One of the options for this type of operation is the REFRESH SCREEN option of the transaction key record. If this field is specified as YES, the window of the previous transaction key will disappear when chaining occurs and the new window is displayed. If REFRESH SCREEN is specified as NO, the first window remains on the screen and the second window is displayed in addition to, and sometimes on top of it.

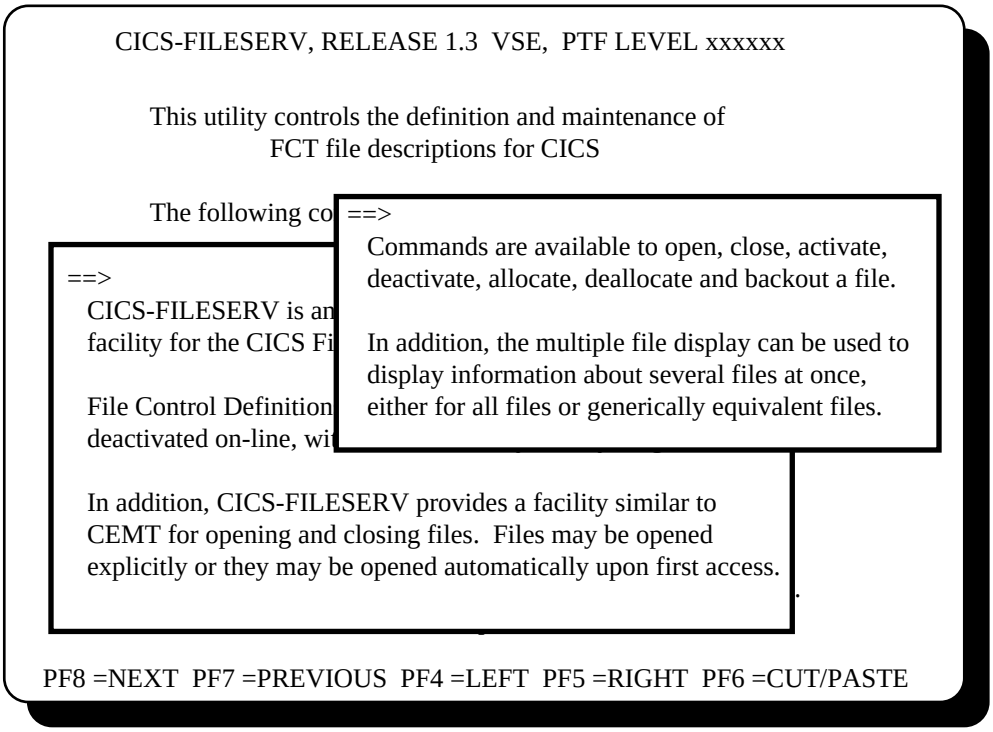
The result is that multiple windows can appear on the screen at one time, each with its associated text. An example of multiple window chaining could appear as follows.

First we have the application screen with a single window displayed:

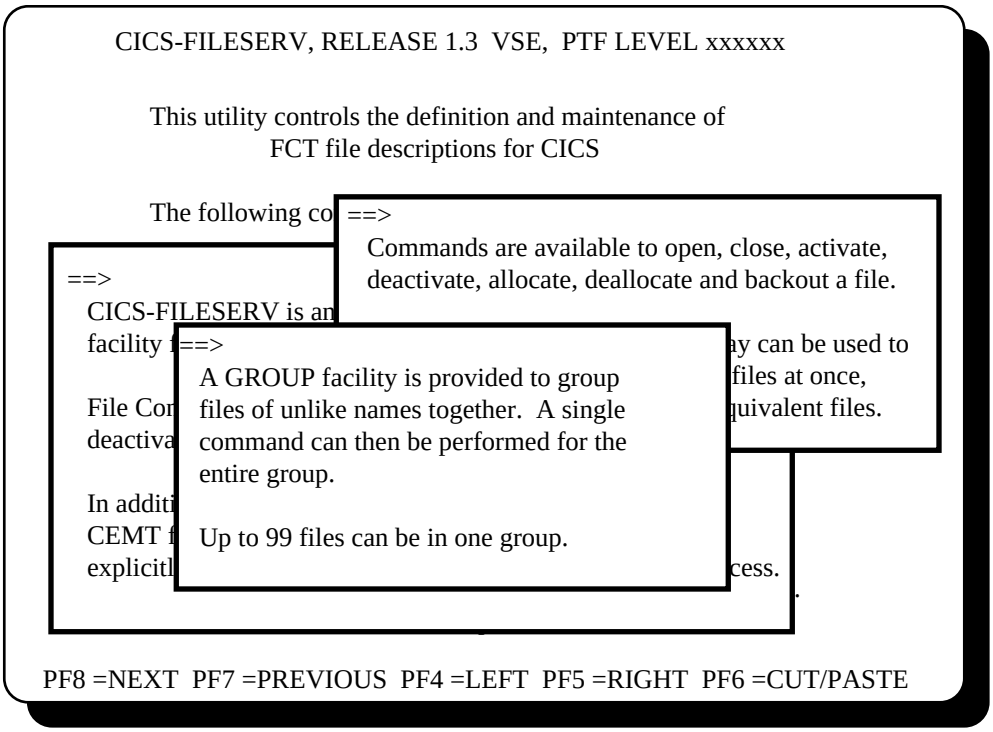

CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx
This utility controls the definition and maintenance of
FCT file descriptions for CICS
The following comands are available:
==>
CICS-FILESERV is an on-line utility which provides an RDO-like facility for the CICS File Control Table.
File Control Definitions can be created, changed, activated and deactivated on-line, without the necessity of recycling CICS.
In addition, CICS-FILESERV provides a facility similar to CEMT for opening and closing
or they may be opened automatically upon first access.
PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Now we browse forward with PF8. If this is the end of text for the first transaction key, and there is another record with the same transaction code, screen ID and field ID but with a different sequence number, you would see still a different display. The next display might appear as follows:


CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx
This utility controls the definition and maintenance of
FCT file descriptions for CICS
The following co
==>
Commands are available to open, close, activate, deactivate, allocate, deallocate and backout a file.
In addition, the multiple file display can be used to display information about several files at once, either for all files or generically equivalent files.
==>
CICS-FILESERV is an
facility for the CICS Fi
File Control Definition
deactivated on-line, wi
In addition, CICS-FILESERV provides a facility similar to CEMT for opening and closing files. Files may be opened explicitly or they may be opened automatically upon first access.
PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

Continuing forward, we might encounter still another transaction key which configures the display as follows:


CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx
This utility controls the definition and maintenance of
FCT file descriptions for CICS
The following co
==>
Commands are available to open, close, activate, deactivate, allocate, deallocate and backout a file.
==>
A GROUP facility is provided to group files of unlike names together. A single command can then be performed for the entire group.
Up to 99 files can be in one group.
PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The window which contains the cursor is the current active window and is the one which responds to any commands or PF key directional options.

Browsing backward from this point with the PREVIOUS command or key will remove the windows one at a time as the beginning of text for each window is passed.

Also, pressing PF3 will revert back to the previous window. Continued pressing of the help key will take you all the way back to the application display with no windows. Pressing the CLEAR key will take you directly back to the application screen, no matter how many windows are on the screen.

HELP DISPLAYS WITHIN HELP TEXT (NESTED HELP)

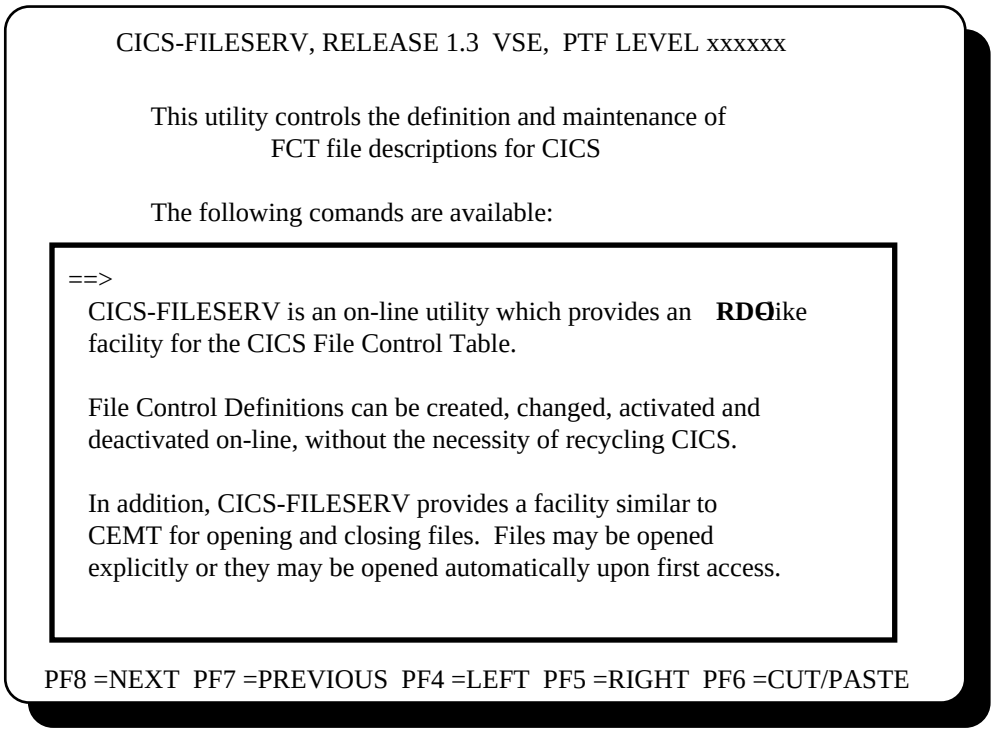
It is possible to define a help screen or help window, then define another help display to be keyed to one of the words of text in the first display. This is called 'nested' help.

The definition for this is accomplished by coding the word in the first display with an unprotected attribute so that the operator can tab to it (it should probably also be high intensity or a different color so as to be noticed), then defining a transaction key using that word as the field text identifier. For this definition, CURSOR IN FIELD TEXT should be specified as YES.

To retrieve the nested help display while in help mode, tab to the field, then press the ENTER key instead of the help key.

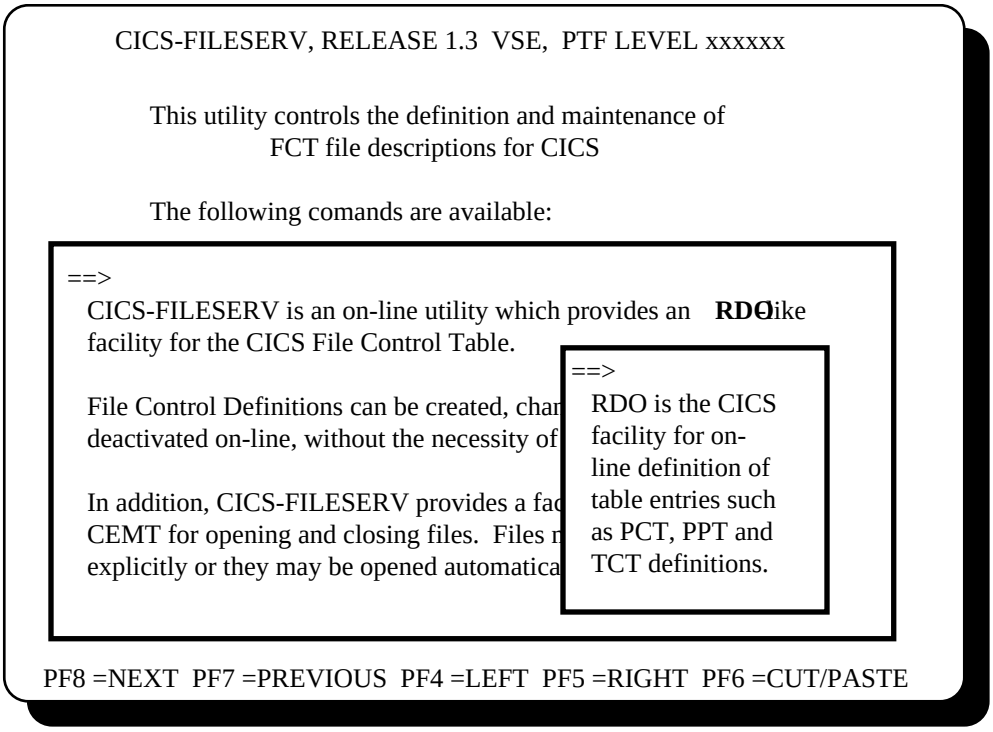
An example of this would be as follows. On the first display, the word RDO might be set up with a transaction key using 'RDO' as a field text identifier. RDO is coded to be high-intensity, unprotected.

The appears as follows:


CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx
This utility controls the definition and maintenance of
FCT file descriptions for CICS
The following comands are available:
==>
CICS-FILESERV is an on-line utility which provides an RDO-like
facility for the CICS File Control Table.
File Control Definitions can be created, changed, activated and
deactivated on-line, without the necessity of recycling CICS.
In addition, CICS-FILESERV provides a facility similar to
CEMT for opening and closing files. Files may be opened
explicitly or they may be opened automatically upon first access.
PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If the operator tabs to the RDO field and presses ENTER, the next display could appear as:


CICS-FILESERV, RELEASE 1.3 VSE, PTF LEVEL xxxxxx
This utility controls the definition and maintenance of
FCT file descriptions for CICS
The following comands are available:
==>
CICS-FILESERV is an on-line utility which provides an RDO-like
facility for the CICS File Control Table.
File Control Definitions can be created, char
deactivated on-line, without the necessity of
In addition, CICS-FILESERV provides a fac
CEMT for opening and closing files. Files r
explicitly or they may be opened automatica
==>
RDO is the CICS facility for on-line definition of table entries such as PCT, PPT and TCT definitions.
PF8 =NEXT PF7 =PREVIOUS PF4 =LEFT PF5 =RIGHT PF6 =CUT/PASTE

The nested window could, of course appear anywhere on the screen, or it could be a full-screen display.

Nested help windows work in the same manner as chained windows. That is, pressing PF3 causes the current window to disappear and reverts back to the previous window in the nest.

EXITING FROM A HELP DISPLAY

When you are ready to exit a help display and return to transaction operation, there are two methods of doing so:

1) Pressing PF3 will remove the current window or help screen and return to the previous level. If only one window or screen is present, this will return you to the application screen.

If multiple help displays are present, either because of screen chaining or nested help displays, pressing PF3 takes you back to the previous display. Pressing it again takes you back another level, etc.

2) Pressing the CLEAR key always returns you to the application screen, regardless of how many help displays are concurrently active.

Once you return to the application screen you may continue with the next input to that application, just as if you had not interrupted it. The cursor is replaced in whatever position it was in when you first pressed the help key and all attributes are restored to their original value.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USING THE CUT/PASTE FEATURE

The cut and paste function is a special feature of HELP-WINDOWS which allows the operator to copy text from the help display into the application screen.

A good use of this feature is when the text in the help window describes all of the valid codes for a given field on the application screen. The operator can place the cursor in the field, press the help key to see all of the valid codes, then, rather than exiting help mode and keying the code in the field, simply place the cursor on the desired code and press the cut/paste key. This will cause the word where the cursor was placed to be copied from the help text and placed into the application field where the cursor was positioned when the help key was originally pressed.

When the word is pasted into the application field, the modified-data-tag for the field is turned on, as it would be if the operator had keyed the word. Thus, when ENTER or a PF key is pressed to the application, the pasted word will transmit to the application program just as if it were keyed.

When defining a help display, you do not need to do anything to enable the cut/paste feature to be used. It will work with any help display. It may be helpful, however, to place unprotected attributes before each word or value which would logically be eligible for a cut/paste operation, thereby allowing the operator to use the TAB key to easily position to the desired entry.

The following limitations and rules apply to the cut/paste feature of HELP-WINDOWS:

1) You can only paste into an unprotected field of the application display.

2) Only one word will be copied when the cut/paste key is pressed. The cursor may be anywhere within that word. If multiple-word copy is desired you must connect the words with a hyphen or underscore or some other character so that HELP-WINDOWS will consider it to be all one word.

3) If the word being pasted is longer than the receiving field, characters will be truncated on the right until the word fits the field.

4) If the word being pasted is shorter than the receiving field, no additional characters are padded on the end.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

(PAGE INTENTIONALLY LEFT BLANK)

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 8. THE MENU GENERATION FEATURE

An optionally licensed feature of CICS-WINDOWS is a facility that enables users to create menus to be used by operators. A menu can be configured such that it can invoke transactions, programs and/or other menus.

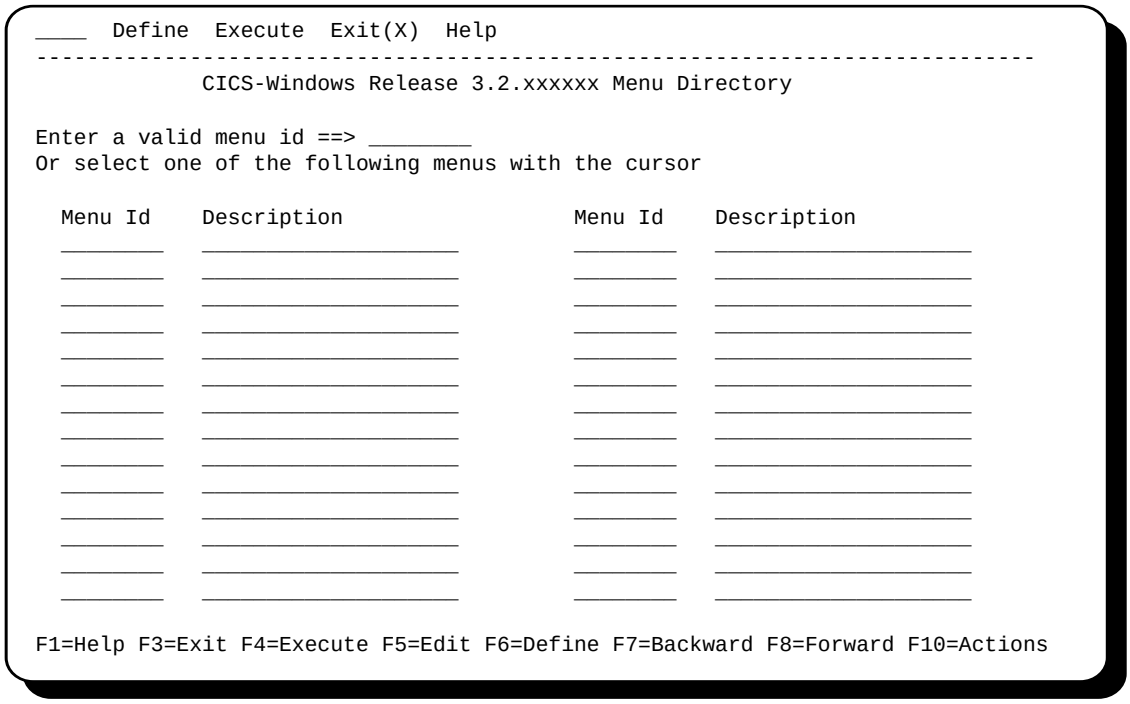
A demonstration menu system is distributed on the installation tape. This system provides examples of methods of coding menus and operation of menus. To invoke the demonstration menu, from a clear screen enter WMNU,DEMOMENU.

Menu generation is accessed through the transaction code - WMNU, which is explained in more detail in the following discussion.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MENU DIRECTORY DISPLAY

To invoke menu generation, enter the transaction code WMNU. Upon pressing ENTER, the Menu Directory display will appear as follows:


____ Define Execute Exit(X) Help
CICS-Windows Release 3.2.xxxxxx Menu Directory
Enter a valid menu id ==> _____
Or select one of the following menus with the cursor
Menu Id Description Menu Id Description
F1=Help F3=Exit F4=Execute F5=Edit F6=Define F7=Backward F8=Forward F10=Actions

FIELDS OF THE MENU DIRECTORY DISPLAY

==> This is the menu ID field and is used to access a menu that may or may not be displayed on this screen. This field is used by keying the menu ID, then pressing the desired PF key; or tabbing the cursor to the function in the function key area, and pressing ENTER.

The columns of fields below this point make up the directory of available menus. All menus on file are displayed in alphabetic order. Each column entry contains two fields:

DESCRIPTION This is a brief description of the menu.

MENU ID This is the name of the menu. Names are assigned to each menu definition for ease of identification. These fields are unprotected so that you may tab to the desired menu then press the PF key of the function that you wish to invoke. You can not change a menu name with this field.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE MENU DIRECTORY DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may select the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP	Display a Help screen for information pertaining to the directory.
PF3 EXIT	Exit the Menu Generation transaction.
PF4 EXECUTE	This is used to execute (invoke) the specified menu.
PF5 EDIT	This is used to view and optionally modify the actual text of the specified menu.
PF6 DEFINE	Display the Menu Definition of the specified menu.
PF7 BACKWARD	Browse backward to the previous page of menu entries (if any).
PF8 FORWARD	Browse forward to the next page of menu entries (if any).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DEFINING AND MAINTAINING MENUS

ADDING A NEW MENU

To add a new menu, from the Menu Definition screen, activate the NEW pull-down menu, then select either ADD or COPY. ADD will create a new menu with all fields blank. COPY will create a new menu with the same specifications and menu text as the currently displayed menu.

Once ADD or COPY is selected, the screen will display the new menu definition, requesting the menu ID that is to be assigned to the new definition. Key a one to eight digit Menu ID and press ENTER to create the new menu definition.

Once created, you may code the fields as described in THE MENU DEFINITION DISPLAY, later in this section.

CHANGING AN EXISTING MENU

To change an already existing menu definition, from the Menu Directory display, select the menu definition that you wish to change, and press the DEFINE key.

The menu definition will then be displayed. To alter the definition, key the changes and press ENTER.

DELETING A MENU

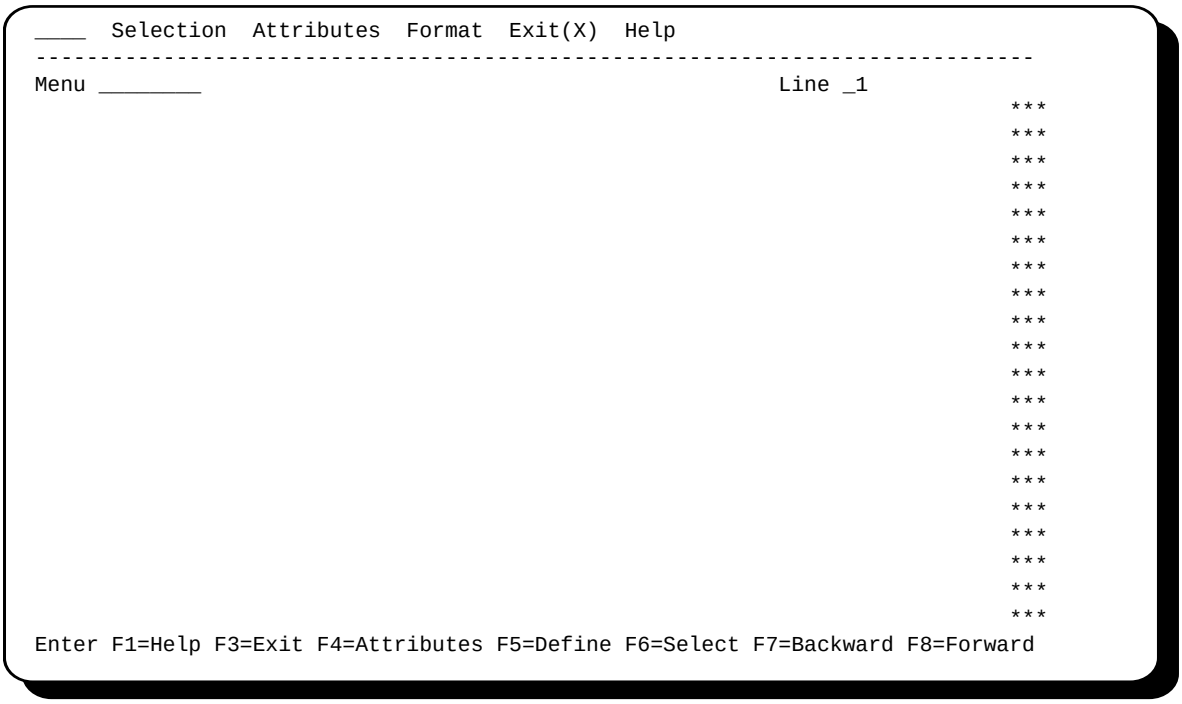
To delete a menu definition, from the Menu Directory display, select the menu definition that you wish to delete, and press the DEFINE key.

The menu definition will then be displayed. To delete the definition, activate the DELETE pull-down menu, and select DELETE.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MENU TEXT EDITOR

This is used to 'paint the screen' the way you want the menu to be displayed. This screen appears when attempting to define or modify a menu by pressing the EDIT key from the Menu Directory or Menu Definition displays. This screen appears as follows:


____ Selection Attributes Format Exit(X) Help
Menu _____ Line _1
***
Enter F1=Help F3=Exit F4=Attributes F5=Define F6=Select F7=Backward F8=Forward

USING THE MENU TEXT EDITOR

To enter text, simply key lines of data, formatting it as desired so far as line spacing, indentation, etc. are concerned. If changes are needed after keying some text, you can simply move the cursor to the area to be changed and type over any existing data. You can use the DEL key to delete characters in a line. To insert characters on a line, position the cursor where the insertion is to be made, press the INS key and type the desired characters (note that you may need to turn NULLS on, which is explained later in this topic). The ERASE EOF key may be used to erase all or part of a line at any time.

When all text for this page has been keyed, press ENTER. This will write the updated text record to file. If the text being entered cannot fit on one screen, you may press the FORWARD key (or change the line number at the top of the display).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

LINE DELETION

To delete an entire line, TAB to the COMMAND FIELD at the end of the line (where the asterisks are) and enter a 'D', then press ENTER. The entire line will be deleted and all following lines will shift up one line.

To delete more than one line at a time, enter a 'D' in more than one COMMAND FIELD before pressing ENTER or enter 'Dnn' on one or more COMMAND FIELDS, where nn is the number of consecutive lines (including the current line) to be deleted.

You may also perform block deletion by placing 'DD' on the first line to be deleted and another 'DD' on the last line to be deleted.

LINE INSERTION

To insert a line between two text lines, tab to the COMMAND FIELD of the preceding line and enter an 'I' over any one of the asterisks, then press ENTER. This will cause a blank line to be inserted following that line, and all subsequent lines will shift down one line. You may then key any desired text on that line.

To insert multiple lines, enter 'Inn' in the preceding COMMAND FIELD, where nn is the number of lines to insert, then press ENTER. This will cause nn blank lines to be inserted after the current line, and all subsequent lines will shift down nn lines.

MOVING AND COPYING TEXT LINES

One or more consecutive text lines can be moved or copied to another location. To move text lines, enter 'M' or 'Mnn' (nn = number of lines to move) in a COMMAND FIELD and press ENTER. This causes the specified line(s) to be deleted from that location. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER. The moved lines will be inserted following that line if the 'A' command is used, or before that line for the 'B' command.

To copy text lines, enter 'C' or 'Cnn' (nn = number of lines to copy) in a COMMAND FIELD and press ENTER. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER.

You may also perform block moves and copies by placing 'MM' or 'CC' on the first line to be moved or copied and another 'MM' or 'CC' on the last line to be moved/copied.

You may enter more than one 'M' or 'C' command on one edit screen, if desired. The result is that all selected lines are grouped together consecutively and inserted as one set wherever the 'A' or 'B' command is placed. For instance, if 'C2' is entered in the command area at line 3, then 'C4' is entered in the command area at line 16, all six text lines will be grouped together and inserted as one set following the line where the 'A' command is placed.

Note that you may use the 'A' or 'B' sub-commands as many times as desired, without performing another 'M' or 'C' sub-command. Each time it is used, the same data that was moved or copied last will be inserted at the new location.

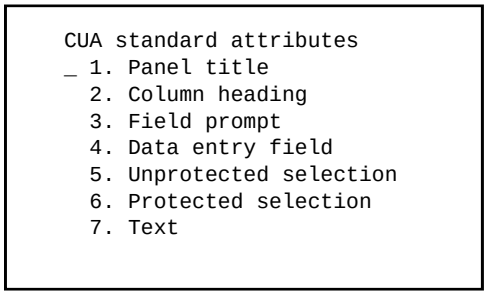
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Unless otherwise specified, all words on the menu will be displayed in low intensity protected. In order to emphasize certain words or lines, you can use the ATTRIBUTES function.

To invoke the Attribute function, press the ATTRIBUTES key (shown in the PF key prompt area). This places the terminal in SET COLOR mode. Note that upon invoking the attribute function, the function key area changes to include additional functions. These additional functions are as follows:

ENTER	If the cursor is inside the attribute window, pressing ENTER will move the cursor to the current attribute pointer position.
F2=SHOW	Show all attributes on the screen as an at-sign (@).
F3=EXIT	Exit Set Attribute mode to the editor.
F4=SET	Set an attribute (as specified in the attribute window) at the current cursor position.
F5=SELECT	Select the attribute at the position of the cursor and display the type of attribute in the attribute window. (If the cursor is inside the attribute window, this will move the window out of the way.)
F6=REMOVE	Remove (delete) the attribute at the position of the cursor.
F7=BWD	Move the current attribute pointer to the previous attribute.
F8=FWD	Move the current attribute pointer to the next attribute.
F9=EXTEND	Toggle the attribute window between CUA and extended attributes.

When the attributes function is invoked, a pop-up window appears. The attributes window can appear in two forms:



- A CUA standards attribute window. This window contains the CUA attributes for creating a display. The following chart portrays the display attributes for the various selections:

Selection	Extended color	Intensity	Protect/unprotect
Panel title	Yellow	High	Protected
Column Heading	Turquoise	High	Protected
Field Prompt	Turquoise	Normal	Protected
Data Entry	Green	Normal	Unprotected
Unprotected Selection	White	Normal	Unprotected
Protected Selection	White	Normal	Protected
Text	Blue	Normal	Protected

- An extended attribute window. This window offers all attribute combinations.

If the CUA standard window is displayed and you wish to use the extended attribute window, press the EXTEND key (shown in the PF key area). This key works as a toggle between the two types of attribute windows.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Upon pressing the ATTRIBUTES key, either a field mark or an asterisk with an overscore will display somewhere on the screen. This is the Current Attribute Pointer. If an attribute is present at that position of the screen, it will appear as a field mark, else it will appear as the overscored asterisk. You may press the FORWARD or BACKWARD keys to position the pointer to the next or previous attributes on the screen, or you may position the cursor to a suspected location of an attribute and press the SELECT key to move the attribute pointer directly to that position.

To set an attribute, simply select the choices within the attribute window by entering the number of the choice. Then position the cursor to a space before the word (or line) that is to contain the attribute features. Then press the SET key (shown in the function key area). The attribute will remain in effect until another attribute or the end of the line is encountered.

To remove an attribute, position the cursor to the attribute to be removed and press the REMOVE key (shown in the PF key area).

[Note] Setting color with either attributes windows may cause extended attributes to be generated in the screen display when viewed. You must be using a terminal that supports extended attributes and the COLOR and/or HILIGHT feature(s) must be set in the CICS terminal control table (TCT) entry (or EXTENDEDDS with RDO) in order to view them. If you view a display on a terminal that does not support extended datastreams, the extended color and highlighting attributes will be removed (in favor of regular field attributes) when the screen is displayed.

THE SELECT WINDOW

While in the Menu Editor, you may view and alter the selection information of a Menu field on the screen by moving the cursor to the field and pressing the SELECT key (shown in the PF key prompt area).

Upon pressing the SELECT key, a popup window will appear showing the selection information of the field. To view the selection information of another field, you may press the FORWARD or BACKWARD keys, or change the SELECTION NUMBER in the popup window and press ENTER.

The information shown in the popup window directly corresponds to the information shown on the Menu Definition Display. For a more detailed description of these fields, see *THE MENU DEFINITION DISPLAY*, earlier in this section.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE MENU TEXT EDITOR

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key.

ENTER — Update the menu or message and/or apply any commands in the command field (to the right of the screen).

PF1 HELP — Display a Help screen for information pertaining to the text editor.

PF3 EXIT — Exit to the Menu Directory display.

PF4 ATTRIBUTES — Set the editor to attribute mode. This will display a different function key area at the bottom of the screen. For more information, please refer to CREATING COLOR AND HIGHLIGHTING under THE MENU TEXT EDITOR, earlier in this section.

PF5 DEFINE — Display the Message Definition screen.

PF6 SELECT — This is used in the Menu Editor to display a popup window with the menu definition selection information about the row that the cursor was in when the SELECT key was pressed. For more information, see THE SELECT WINDOW, earlier in this section.

PF7 BACKWARD — Display the previous page of text.

PF8 FORWARD — Display the next page of text.

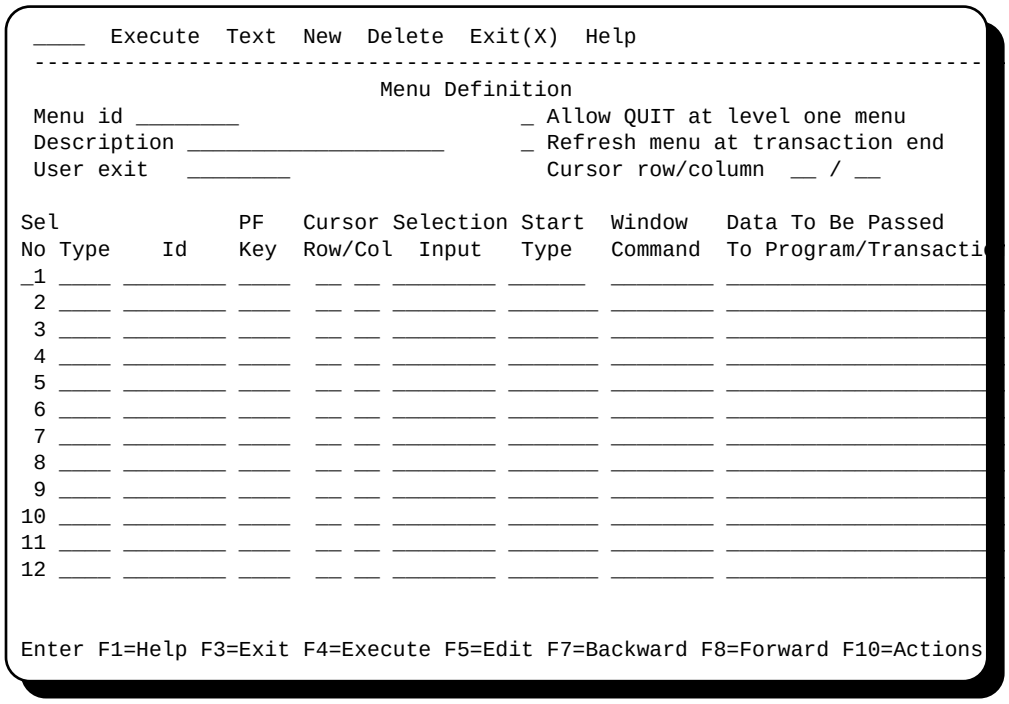
PF9 NULL ON/OFF — The editor can display trailing spaces as spaces or nulls. If spaces are used, you may type text in any column of the editor and upon pressing ENTER, the text will remain where it was typed, however, since all characters in a line are full, the INSERT key will not operate. If nulls are used, the INSERT key will operate properly, but if keyed text is preceded with nulls, when ENTER is pressed the text will shift left one column for every preceding null. This key works as a toggle switch. Pressed once turns nulls on; pressed again turns nulls off.

PF11 SCALE ON/OFF — This is used to display a scale along the top of the editor. This works as a toggle switch to display or remove the scale.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MENU DEFINITION DISPLAY

This screen is used to define each item on the menu and specify what is to happen when that selection is made. This screen appears when defining or modifying parameters of a menu, (by pressing the DEFINE key from the Menu Editor display). Portions of this screen will appear when the SELECT key is pressed in the text editor. This screen appears as follows:


____ Execute Text New Delete Exit(X) Help
Menu Definition
Menu id _____ _ Allow QUIT at level one menu
Description _____ _ Refresh menu at transaction end
User exit _____ Cursor row/column __ / __
Sel No Type Id PF Key Cursor Row/Col Selection Input Start Type Window Command Data To Be Passed To Program/Transacti
Enter F1=Help F3=Exit F4=Execute F5=Edit F7=Backward F8=Forward F10=Actions

FIELDS OF THE MENU DEFINITION DISPLAY

ALLOW QUIT AT LEVEL ONE MENU

If selected, an operator will be allowed to exit from this menu to CICS, if this menu was not entered from another menu.

CURSOR ROW/COL

These fields specify the row and column position of the cursor when the menu is invoked.

CURSOR ROW/COL (on Selection Fields)

These two fields are used to specify a position of the cursor to which the operator can tab and press ENTER, to invoke this function, if cursor selection of menu items is desired.

If desired, the column field can be coded as '00', to specify a generic column. In other words, to select this option, the cursor may appear anywhere within the specified row when ENTER is pressed. If column is coded, the cursor must be in exactly that position for selection to occur.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DATA TO BE PASSED TO PROGRAM/TRANSACTION

This field defines the data to be passed to the application program in the terminal I/O area or Interval Control Element. For ATTACH, the data defined is the data that would be entered immediately following the transaction code if this transaction were entered on a terminal screen. Thus, if only the trancode is required as input, this field should be left blank. For START, the data should be in the format expected by the target program.

Example: To automatically start a transaction that would normally be entered on the screen in the form:

```
FSRV I  ACCTFIL  A*
```

the following coding would be required in the statement (in addition to the method(s) of selection of this option from the menu):

```
TYPE          TRAN
ID            FSRV
START TYPE    ATTACH
DATA          _I_ACCTFIL_A*
```

DESCRIPTION

This is a short description of this menu definition. This field is not mandatory but is recommended for ease of recognition since it will appear on the Menu Directory Display.

ID This is the actual ID of the function that is to be invoked. Valid entries are:

For TYPE=ESC	This field should be left blank.
For TYPE=MENU or SUBM	Code the Menu ID of the CICS-WINDOWS menu to be invoked.
For TYPE=PROG	Code the name of the program to be invoked, as it is known to CICS.
For TYPE=TRAN	Code the name of the transaction to be invoked, as it is known to CICS.
For TYPE=QUIT	This field should be left blank.

MENU ID This is the name of this menu. This name will appear in the Menu Directory display. It is a mandatory field.

PF KEY This field is used to specify a PF key that the operator can press to invoke this selection. This field is not mandatory. Entries are:

CLR	This is the mnemonic for the CLEAR key.
ENTR	This is the mnemonic for the ENTER key.
PFxx	Any valid PF or PA key, coded as PFx, PFxx or PAx.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

REFRESH MENU AT TRANSACTION END

If selected and the operator entered a transaction through this menu, display this menu when that transaction is terminated.

[Note]: Transaction end is recognized when there is no Return Transid and the task is not conversational. If selected, CICS-WINDOWS must be active on the terminal, for this feature to operate properly.

SEL NO This is the selection number. This is used only for displaying records beyond the ninth entry. The top most SEL NO field is unprotected. This enables you to change the starting selection number of the display. For instance, to display the next nine entries, key a "10" in this field and press ENTER.

SELECTION INPUT

This field is used to specify data that the operator may key (in an entry field) in order to invoke this menu option. (i.e. A, B, C... or 1, 2, 3...)

START TYPE This field is used to specify the technique for CICS-WINDOWS to use when starting a program or transaction. Valid entries are:

For TYPE=PROG

LINK This specifies that this program will be started and upon termination, control will return immediately to this menu.

XCTL This specifies that control will be transferred entirely to this program. Upon completion, CICS-WINDOWS will respond according to the coding of the REFRESH MENU AT TRANSACTION END field.

For TYPE=TRAN

ATTACH This means that the transaction will be started in the same way that CICS starts a transaction when the transaction code is entered from the terminal. That is, an I/O area is passed to the application containing the transaction code in the first position and any associated data following.

[Note]: ATTACH must be used if data is to be passed to the application in a terminal I/O area, or if the transaction being started runs in a remote region through MRO or ISC.

START START means that the transaction will be initiated using an EXEC CICS START. When using START, no terminal I/O area is passed to the application program. Data is passed in an interval control record, which the program can retrieve.

[Note]: There is no "preferred" method here. If a transaction does not need incoming data, either method will work. ATTACH will be slightly quicker since it avoids the CICS Automatic Transaction Initiation mechanism. If the program expects to receive data with an EXEC CICS RETRIEVE command, START must be used.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TYPE This field is used to specify the type of selection that this menu option will invoke. Valid entries are:

INPT The data entered in this field will serve as input for another menu option specifying '&INP' in either the 'Id' or 'Data to be Passed to Program/Transaction' fields of the Menu Definition Display.

ESC When selected, this option will exit this menu to a clear screen no matter what level of menu nesting is in effect. Any number of transactions may be performed, and the current menu restored by entering the trancode WMNU.

Using ESC may lead to a recursive situation in which no other menu may be invoked with the WMNU transaction since WMNU restores the transaction at the point the escape option was selected. This feature of ESC may be disabled by entering, from a clear screen, WMNU,$CANCEL. Any menu may then be
invoked in the normal manner. This process will not be necessary if another key is assigned with the QUIT type, allowing the operator to exit the menu to the next higher level.

MENU When selected, this option will invoke another CICS-WINDOWS menu.

PROG When selected, this option will invoke a program.

SUBM When selected, this option will invoke a sub-menu. A sub-menu is a menu that appears within a window. Instead of displaying a completely different menu, the sub-menu window simply overlaps a portion of the screen.

TRAN When selected, this option will invoke a transaction.

QUIT When selected, this option will exit this menu to the next higher level (Either a clear screen or the menu that called this menu, depending on the method in which this menu was invoked.)

spaces This field may also be used to delete a menu selection line simply by erasing the TYPE field with the ERASE EOF key or space bar.

USER EXIT This field is used to specify a user exit program that will be called anytime this menu is displayed. The exit program may temporarily alter the way the menu displays and operates. For instance, if an operator is not authorized to used a certain transaction, the exit program can temporarily delete the option from the menu, so that the operator never even sees the option. For more information on this user exit, see *THE MENU GENERATION FEATURE USER EXIT*, in section 15 - *USER EXITS AND PROGRAM INTERFACES*.

WINDOW COMMAND
This field is used to specify any CICS-WINDOWS API commands to be processed upon selection of this menu. For more information on API commands, see *THE APPLICATION PROGRAM INTERFACE*, in section 15 - *USER EXITS AND PROGRAM INTERFACES*.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE MENU DEFINITION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may activate the desired option by pressing the associated PF key or by tabbing to the option in the function key area and pressing ENTER.

ENTER	Apply any changes or additions made to this screen; or effect the function in the function key area specified by the option to which the cursor pointed when ENTER was pressed.
PF1 HELP	Display a Help screen for information pertaining to this screen or to the field that the cursor pointed when the HELP key was pressed.
PF3 EXIT	Exit to the Menu Directory display.
PF4 EXECUTE	Execute this menu. This will exercise the menu as the operator will see it.
PF5 EDIT	Display the menu editor to view or change the actual menu.
PF7 BACKWARD	Browse backward to the previous menu definition.
PF8 FORWARD	Browse forward to the next menu definition.
	In addition to the PF key functions, the action bar may be used to create a new menu, copy this menu, or delete this menu.

INITIATION AND OPERATION OF MENUS

Two methods for initiating a menu are available. One method is to press the EXECUTE key (shown in the PF key prompt area) from the Menu Directory or Menu Definition displays. The second method is to ENTER the WMNU,xxxxxxxx command from a clear screen (where xxxxxxxx is the Menu ID of the menu to be executed).

Once a menu is operating, the method(s) of selecting menu options is controlled entirely by the menu definition. Any of the following methods may be available:

- Tab to the option and press ENTER.
- Enter a selection in an entry field.
- Press a PF or PA key.
- Press the CLEAR key.

AUTOMATIC INITIATION OF MENUS

Menus can be defined to automatically initiate when a session is first entered, either by toggling into it or switching to an empty window.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

To do this, use the Application Startup Table of the CICS-WINDOWS customization. Code WMNU as the Transaction ID. ATTACH as the Start Type, and a comma followed by the menu ID in the Data field. You may code the same menu to be started in all sessions, or have different menus, as desired.

See *THE APPLICATION STARTUP TABLE*, in section 11 - *CUSTOMIZATION*, for more information.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USING REPLACABLE PARAMETERS IN MENUS

A menu option can be designated to supply data to another menu option by specifying INPT as type for the supplying menu option and employing the replaceable parameter &INP in either the 'Id' or 'Data to be Passed to Program/Transaction' fields of the receiving menu option. For example, if the following Menu Definition is specified:

```
____  Execute  Text  New  Delete  Exit(X)  Help
--------------------------------------------------------------------------------
                            Menu Definition
 Menu id ________                        _ Allow QUIT at level one menu
 Description ____________________        _ Refresh menu at transaction end
 User exit   ________                      Cursor row/column  __ / __

Sel              PF   Cursor Selection Start  Window   Data To Be Passed
No Type    Id    Key  Row/Col  Input   Type   Command  To Program/Transaction
_1 INPT _________ ____  05 22 _________ ______  ________ ____________________________
 2 TRAN &INP____ PF11  __ __ _________ ATTACH_ ________ ____________________________
 3 TRAN CEMT____ PF12  __ __ _________ START__ ________ _SET_PROG_(&INPFILE)_NEW_
 4 ____ _________ ____  __ __ _________ _______ ________ ____________________________
 5 ____ _________ ____  __ __ _________ _______ ________ ____________________________
 6 ____ _________ ____  __ __ _________ _______ ________ ____________________________
 7 ____ _________ ____  __ __ _________ _______ ________ ____________________________
 8 ____ _________ ____  __ __ _________ _______ ________ ____________________________
 9 ____ _________ ____  __ __ _________ _______ ________ ____________________________
10 ____ _________ ____  __ __ _________ _______ ________ ____________________________
11 ____ _________ ____  __ __ _________ _______ ________ ____________________________
12 ____ _________ ____  __ __ _________ _______ ________ ____________________________


Enter F1=Help F3=Exit F4=Execute F5=Edit F7=Backward F8=Forward F10=Actions
```

If the operator enters HWIN in the field specified as Sel No 1, then presses PF11, which selects Sel No 2, the transaction HWIN will be attached, since Sel No 2 contains in its 'Id' field the replaceable parameter &INP. The replaceable parameter is not limited to four characters; &INP is a symbol that is replaced by the specified string.

The third selection field illustrates the use of the replaceable parameter with additional data appended. In this case, the selection will newcopy the HWINFILE, the parameter again being replaced by the INPT string.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

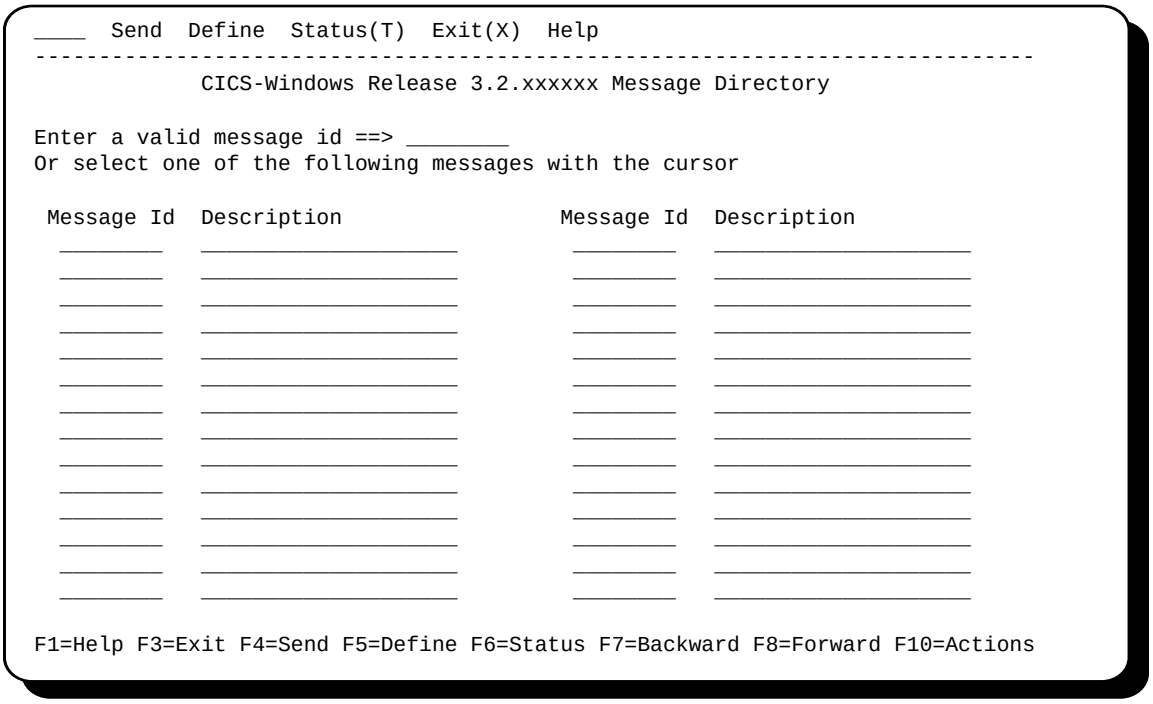
Section 9. THE MESSAGE BROADCASTING FEATURE

An optionally licensed feature of CICS-WINDOWS is the flexible message broadcasting facility that may be used to transmit messages, news, inquiries etc. The message may be transmitted as a full-screen image or a pop-up window. In addition, the sender can specify one of four ways that the message is to be retrieved by the target terminal: immediately, upon response to a prompt, upon pressing the toggle key or upon pressing the CLEAR key. Also, the message may be broadcasted to a specific terminal, a specific user, a list of terminals and/or users, or to all terminals in this CICS.

The message broadcasting facility is accessed through the transaction code - WMSG, which is explained in detail in the following discussion.

THE MESSAGE DIRECTORY DISPLAY

To invoke the message broadcasting transaction, enter the transaction code WMSG. Upon pressing ENTER, the Message Directory display will appear generally as follows:


____ Send Define Status(T) Exit(X) Help
CICS-Windows Release 3.2.xxxxxx Message Directory
Enter a valid message id ==> ________
Or select one of the following messages with the cursor
Message Id Description
Message Id Description
F1=Help F3=Exit F4=Send F5=Define F6=Status F7=Backward F8=Forward F10=Actions

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE MESSAGE DIRECTORY DISPLAY

==> This is the message ID field and is used to access a message that may or may not be displayed on this screen. This field is used by keying the message ID, then pressing the desired PF key; or tabbing the cursor to the function in the function key area, and pressing ENTER.

The columns of fields below this point make up the directory of available messages. All messages on file are displayed in alphabetic order. Each column entry contains two fields:

DESCRIPTION This is a brief description of the message.

MESSAGE ID This is the name of the message. Since CICS-WINDOWS can retain messages along with all associated parameters, names are assigned to each message definition for ease of identification. These fields are unprotected so that you may tab to the desired message then press the PF key of the function that you wish to invoke. You can not change a message name with this field.

ENTERING COMMANDS AT THE MESSAGE DIRECTORY DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP Display a Help screen for information pertaining to the directory.

PF3 EXIT Exit the Message Broadcasting facility.

PF4 SEND This is used to broadcast the message specified by the field that the cursor is in at the time PF4 is pressed or by the name keyed in the ==> field.

PF5 DEFINE This is used to view and optionally modify the message or the parameters of the message specified by the field that the cursor is in at the time PF5 is pressed or by the name keyed in the ==> field.

PF6 STATUS This is used to display the Active Message Status display for the message specified by the field that the cursor is in at the time PF6 is pressed or by the name keyed in the ==> field.

PF7 BACKWARD Browse backward to the previous entry of the message directory.

PF8 FORWARD Browse forward to the next entry of the message directory.

PF10 ACTIONS Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DEFINING AND MAINTAINING MESSAGES

ADDING A NEW MESSAGE

To add a new message, from the Message Definition screen, activate the NEW pull-down menu, then select either ADD or COPY. ADD will create a new message with all fields blank. COPY will create a new message with the same specifications and message text as the currently displayed message .

Once ADD or COPY is selected, the screen will display the new message definition, requesting the message ID that is to be assigned to the new definition. Key a one to eight digit name and press ENTER to create the new message definition.

Once created, you may code the fields as described in *THE MESSAGE DEFINITION DISPLAY*, later in this section.

CHANGING AN EXISTING MESSAGE

To change an already existing message definition, from the Message Directory display, select the message definition that you wish to change, and press the DEFINE key.

The message definition will then be displayed. To alter the definition, key the changes and press ENTER.

DELETING A MESSAGE

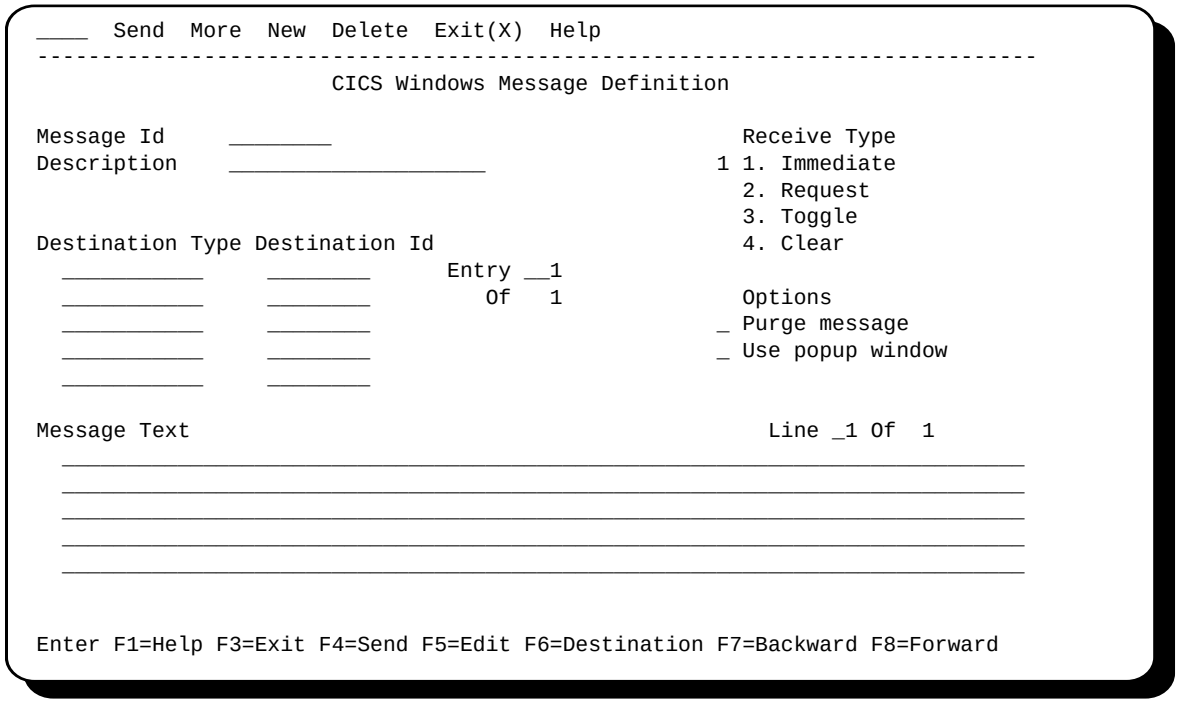
To delete a message definition, from the Message Directory display, select the message definition that you wish to delete, and press the DEFINE key.

The message definition will then be displayed. To delete the definition, activate the DELETE pull-down menu, and select DELETE.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MESSAGE DEFINITION DISPLAY

This screen is used to specify the broadcast parameters of a message. This screen is accessed by pressing the DEFINE key from the Message Directory display and appears generally as follows:



FIELDS OF THE MESSAGE DEFINITION DISPLAY

DESCRIPTION This is a short description of this message definition. This field is not mandatory but is recommended for ease of recognition.

DESTINATION ID This specifies the terminal or operator ID of the target.

For DESTINATION TYPE=TERMINAL

Code the terminal ID, as it is known to CICS. This may be coded in any of the following ways:

1) Code an individual statement for each terminal to be defined, coding this field as the four-character terminal ID of the terminal to be configured.

2) Define a generic definition using "wild-card" characters. Code this operand with question marks (?) in one or more positions of the terminal ID. With this method, the message is broadcast to any terminal that all positions in the terminal ID match the corresponding positions in this field where question marks are not coded.

3) Define a single statement which applies to all terminals. Code this field as ????. With this method, this statement applies to every terminal in this CICS.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

For DESTINATION TYPE=OPERATOR ID

Code the operator name, as it appears in the TCT. This is the three-character CICS operator ID of the operator. This may be coded in any of the following ways:

1) Code an individual statement for each operator, coding the three-character operator ID of the user to receive the message.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the operator ID. With this method, the message is broadcast to any terminal that all positions in the operator ID match the corresponding positions where question marks are not coded.

3) Define a single statement which applies to all users. Code the field as ????. With this method, the message is broadcast to every operator in this CICS.

For DESTINATION TYPE=USER ID

Code the user ID, as it appears in the SNT. This is the eight-character user ID of the operator. This may be coded in any of the following ways:

1) Code an individual statement for each operator, coding the eight-character user ID of the user to receive the message.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the user ID. With this method, the message is broadcast to any terminal that all positions in the user ID match the corresponding positions where question marks are not coded.

3) Define a single statement that applies to all users. Code the field as ????????. With this method, the message is broadcast to every operator in this CICS.

DESTINATION TYPE
This specifies the targeting method to be used. Valid entries are:

TERMINAL Code 'TERMINAL' or 'T' if this entry is to specify a terminal as a target.

OPERATOR ID
Code 'OPERATOR ID' or 'O' if this entry is to specify an operator as a target.

USER ID Code 'USER ID' or 'U' if this entry is to specify a user as a target. Note that this is not valid for CICS releases prior to 1.7.

[Note]: If multiple entries point to the same physical terminal, all but the first entry will be ignored. (i.e. If a TERMINAL and OPERATOR ID entry is coded, and both entries point to the same terminal, CICS-WINDOWS will only honor the first entry.)

ENTRY xxx OF yyy
If the all message destination entries will not display on five lines, the xxx in this field may be altered to reposition the text to display starting with the specified line.

Note that you may also press the DESTINATION key (shown in the PF key prompt area). This will cause the display to change so that the entire screen is used to display the destination entries.

LINE xx OF yy If the entire message does not display on five lines, the xx in this field may be altered to reposition the text to display starting with the specified line.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MESSAGE ID This is the name of this message. This name will appear in the Message Directory display and is mandatory.

MESSAGE TEXT This is the actual text of the message. If desired, you may use these lines for creating or altering the message (instead of the message editor).

OPTIONS These are special options that apply to all message and receive types.
Valid entries are:

PURGE MESSAGE
If selected, this specifies that this message definition is to be deleted, once all targeted terminals have viewed the message. If this option is not selected, this message definition will not be deleted, and will be available for future use.

USE POPUP WINDOW
If selected, a pop-up window will appear on the screen of the target terminal(s) to display the message, instead of using the full screen to display the message.

RECEIVE TYPE This is the method of message retrieval for the target terminal. Valid selections are:

IMMEDIATE The current display on the terminal is saved and the message is displayed. When the operator presses CLEAR, the original display is restored.

REQUEST The operator must enter the WMSG transaction to retrieve any messages. This makes the message broadcaster function similar to an electronic mail package. The message may also be requested from the Control Window.

TOGGLE The message will be displayed the next time the operator presses any toggle key or performs any function that causes a session switch. When CLEAR is subsequently pressed, the session switch will be completed.

CLEAR When the operator of the target terminal presses the CLEAR key, the message will display. When CLEAR is pressed again, CICS-WINDOWS will honor the clear, and act accordingly.

ENTERING COMMANDS AT THE MESSAGE DEFINITION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes or additions made to this screen, pull down the specified menu from the Action Bar, or effect the specified function in the PF key prompt line. Note that when pulling-down a menu or effecting a function, selection is specified by the position of the cursor when ENTER is pressed.

PF1 HELP Display a Help screen for information pertaining to this screen.

PF3 EXIT Exit to the Message Directory display.

PF4 SEND This is used to broadcast the currently displayed message.

PF5 EDIT This is used to view and optionally modify the message text of this message definition.

PF6 DESTINATION This displays a screen to edit as many as 56 destination specifications.

PF7 BACKWARD Browse backward to the previous message definition.

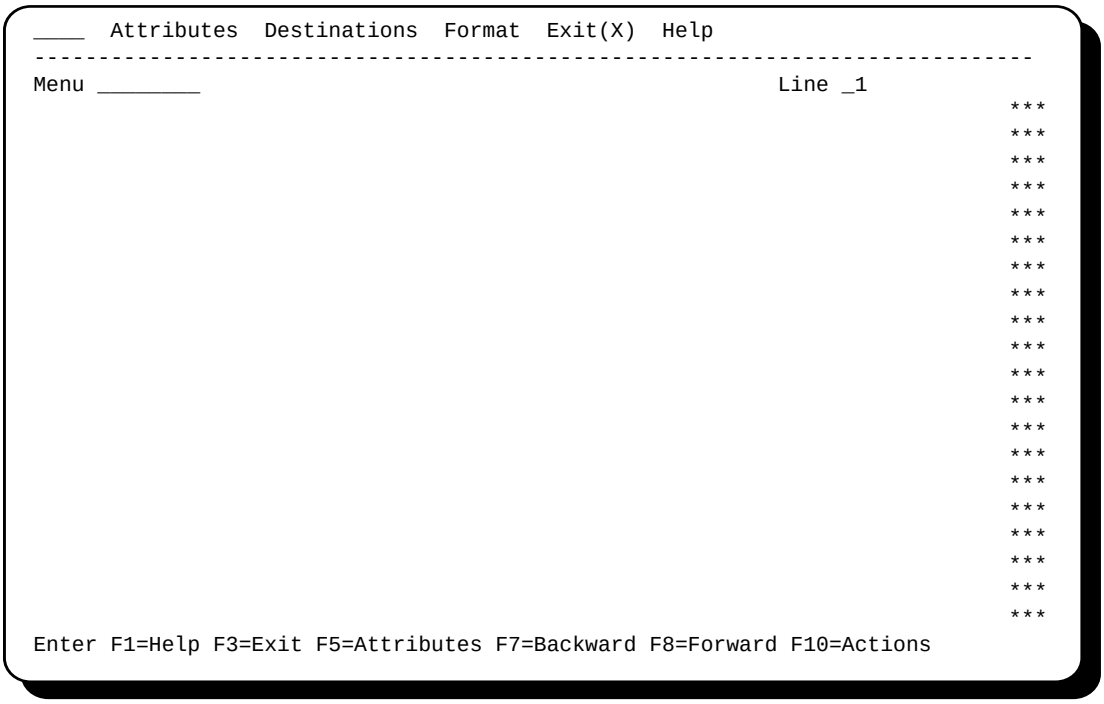
© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PF8 FORWARD Browse forward to the next message definition.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MESSAGE TEXT EDITOR

This is used to 'paint the screen' the way you want the message to be displayed. This screen appears when attempting to define or modify a message by pressing the EDIT key from the Message Directory or Message Definition displays. This screen appears as follows:


____ Attributes Destinations Format Exit(X) Help
Menu _____ Line _1
Enter F1=Help F3=Exit F5=Attributes F7=Backward F8=Forward F10=Actions

USING THE MESSAGE TEXT EDITOR

To enter text, simply key lines of data, formatting it as desired so far as line spacing, indentation, etc. are concerned. If changes are needed after keying some text, you can simply move the cursor to the area to be changed and type over any existing data. You can use the DEL key to delete characters in a line. To insert characters on a line, position the cursor where the insertion is to be made, press the INS key and type the desired characters (note that you may need to turn NULLS on, which is explained later in this topic). The ERASE EOF key may be used to erase all or part of a line at any time.

When all text for this page has been keyed, press ENTER. This will write the updated text record to file. If the text being entered cannot fit on one screen, you may press the FORWARD key (or change the line number at the top of the display).

LINE DELETION

To delete an entire line, TAB to the COMMAND FIELD at the end of the line (where the asterisks are) and enter a 'D', then press ENTER. The entire line will be deleted and all following lines will shift up one line.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

To delete more than one line at a time, enter a 'D' in more than one COMMAND FIELD before pressing ENTER or enter 'Dnn' on one or more COMMAND FIELDS, where nn is the number of consecutive lines (including the current line) to be deleted.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

You may also perform block deletion by placing 'DD' on the first line to be deleted and another 'DD' on the last line to be deleted.

LINE INSERTION

To insert a line between two text lines, tab to the COMMAND FIELD of the preceding line and enter an 'I' over any one of the asterisks, then press ENTER. This will cause a blank line to be inserted following that line, and all subsequent lines will shift down one line. You may then key any desired text on that line.

To insert multiple lines, enter 'Inn' in the preceding COMMAND FIELD, where nn is the number of lines to insert, then press ENTER. This will cause nn blank lines to be inserted after the current line, and all subsequent lines will shift down nn lines.

MOVING AND COPYING TEXT LINES

One or more consecutive text lines can be moved or copied to another location. To move text lines, enter 'M' or 'Mnn' (nn = number of lines to move) in a COMMAND FIELD and press ENTER. This causes the specified line(s) to be deleted from that location. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER. The moved lines will be inserted following that line if the 'A' command is used, or before that line for the 'B' command.

To copy text lines, enter 'C' or 'Cnn' (nn = number of lines to copy) in a COMMAND FIELD and press ENTER. Now place an 'A' (insert-after) in the COMMAND FIELD on the line preceding the new location, or a 'B' (insert-before) on the line following the new location and press ENTER.

You may also perform block moves and copies by placing 'MM' or 'CC' on the first line to be moved or copied and another 'MM' or 'CC' on the last line to be moved/copied.

You may enter more than one 'M' or 'C' command on one edit screen, if desired. The result is that all selected lines are grouped together consecutively and inserted as one set wherever the 'A' or 'B' command is placed. For instance, if 'C2' is entered in the command area at line 3, then 'C4' is entered in the command area at line 16, all six text lines will be grouped together and inserted as one set following the line where the 'A' command is placed.

Note that you may use the 'A' or 'B' sub-commands as many times as desired, without performing another 'M' or 'C' sub-command. Each time it is used, the same data that was moved or copied last will be inserted at the new location.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Unless otherwise specified, all words in the message will be displayed in low intensity protected. In order to emphasize certain words or lines, you can use the ATTRIBUTES function.

To invoke the Attribute function, press the ATTRIBUTES key (shown in the PF key prompt area). This places the terminal in SET COLOR mode. Note that upon invoking the attribute function, the function key area changes to include additional functions. These additional functions are as follows:

ENTER	If the cursor is inside the attribute window, pressing ENTER will move the cursor to the current attribute pointer position.
F2=SHOW	Show all attributes on the screen as an at-sign (@).
F3=EXIT	Exit Set Attribute mode to the editor.
F4=SET	Set an attribute (as specified in the attribute window) at the current cursor position.
F5=SELECT	Select the attribute at the position of the cursor and display the type of attribute in the attribute window. (If the cursor is inside the attribute window, this will move the window out of the way.)
F6=REMOVE	Remove (delete) the attribute at the position of the cursor.
F7=BWD	Move the current attribute pointer to the previous attribute.
F8=FWD	Move the current attribute pointer to the next attribute.
F9=EXTEND	Toggle the attribute window between CUA and extended attributes.

When the attributes function is invoked, a pop-up window appears. The attributes window can appear in two forms:

```
CUA standard attributes
_ 1. Panel title
  2. Column heading
  3. Field prompt
  4. Data entry field
  5. Unprotected selection
  6. Protected selection
  7. Text
```

```
   Color           Protect
_ 1. Blue       _ 1. Protect
  2. Red          2. Unprotect
  3. Pink
  4. Green        Highlight
  5. Turquoise  _ 1. Normal
  6. Yellow       2. Blink
  7. White        3. Reverse
  8. Normal       4. Underscore
```

- A CUA standards attribute window. This window contains the CUA attributes for creating a display. The following chart portrays the display attributes for the various selections:

Selection	Extended color	Intensity	Protect/unprotect
Panel title	Yellow	High	Protected
Column Heading	Turquoise	High	Protected
Field Prompt	Turquoise	Normal	Protected
Data Entry	Green	Normal	Unprotected
Unprotected Selection	White	Normal	Unprotected
Protected Selection	White	Normal	Protected
Text	Blue	Normal	Protected

- An extended attribute window. This window offers all attribute combinations.

If the CUA standard window is displayed and you wish to use the extended attribute window, press the EXTEND key (shown in the PF key area). This key works as a toggle between the two types of attribute windows.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Upon pressing the ATTRIBUTES key, either a field mark or an asterisk with an overscore will display somewhere on the screen. This is the Current Attribute Pointer. If an attribute is present at that position of the screen, it will appear as a field mark, else it will appear as the overscored asterisk. You may press the FORWARD or BACKWARD keys to position the pointer to the next or previous attributes on the screen, or you may position the cursor to a suspected location of an attribute and press the SELECT key to move the attribute pointer directly to that position.

To set an attribute, simply select the choices within the attribute window by entering the number of the choice. Then position the cursor to a space before the word (or line) that is to contain the attribute features. Then press the SET key (shown in the function key area). The attribute will remain in effect until another attribute or the end of the line is encountered.

To remove an attribute, position the cursor to the attribute to be removed and press the REMOVE key (shown in the PF key area).

[Note]: Setting color with either attributes windows may cause extended attributes to be generated in the screen display when viewed. You must be using a terminal that supports extended attributes and the COLOR and/or HILIGHT feature(s) must be set in the CICS terminal control table (TCT) entry (or EXTENDEDDS with RDO) in order to view them. If you view a display on a terminal that does not support extended datastreams, the extended color and highlighting attributes will be removed (in favor of regular field attributes) when the screen is displayed.

ENTERING COMMANDS AT THE MESSAGE TEXT EDITOR

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key.

ENTER — Update the message or message and/or apply any commands in the command field (to the right of the screen).

PF1 HELP — Display a Help screen for information pertaining to the text editor.

PF3 EXIT — Exit to the Message Directory display.

PF5 ATTRIBUTES

Set the editor to attribute mode. This will display a different function key area at the bottom of the screen. For more information, please refer to *CREATING COLOR AND HIGHLIGHTING* under *USING THE MESSAGE AND MESSAGE TEXT EDITOR*, earlier in this section.

PF7 BACKWARD — Display the previous page of text.

PF8 FORWARD — Display the next page of text.

PF9 NULL ON/OFF

The editor can display trailing spaces as spaces or nulls. If spaces are used, you may type text in any column of the editor and upon pressing ENTER, the text will remain where it was typed, however, since all characters in a line are full, the INSERT key will not operate. If nulls are used, the INSERT key will operate properly, but if keyed text is preceded with nulls, when ENTER is pressed the text will shift left one column for every preceding null. This key works as a toggle switch. Pressed once turns nulls on; pressed again turns nulls off.

PF11 SCALE ON/OFF

This is used to display a scale along the top of the editor. This works as a toggle switch to display or remove the scale.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE ACTIVE MESSAGE STATUS DISPLAY

The Active Message Status screen displays a summary of the active messages (messages that have been sent during the period that CICS has been active). The Active Message Status screen is displayed by pressing the STATUS key from the Message Directory display. It appears generally as follows:

```
____  Define  Status  Exit(X)  Help
--------------------------------------------------------------------------------
                            Active Message Status

Enter a message number ==>  ____ ________
Or select one of the following messages with the cursor

  Message Id       From      Time     Date      Sent  Rcvd  Purged
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO
 ____ _________   ____ ____ HH:MM:SS   YY.DDD       0     0    NO

F1=Help F3=Exit F5=Define F6=Status F7=Backward F8=Forward F10=Actions
```

FIELDS OF THE ACTIVE MESSAGE STATUS DISPLAY

DATE	This is the date that this message was sent.
FROM	This is the terminal ID and operator ID of the terminal from which this message was sent.
MESSAGE ID	This is the name of the message.
PURGED	This indicates whether this message definition has been deleted.
RCVD	This is the number of targeted terminals that have received and responded to this message.
SENT	This is the number of terminals that have been targeted to receive this message.
TIME	This is the time that the message was sent.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE ACTIVE MESSAGE STATUS DISPLAY

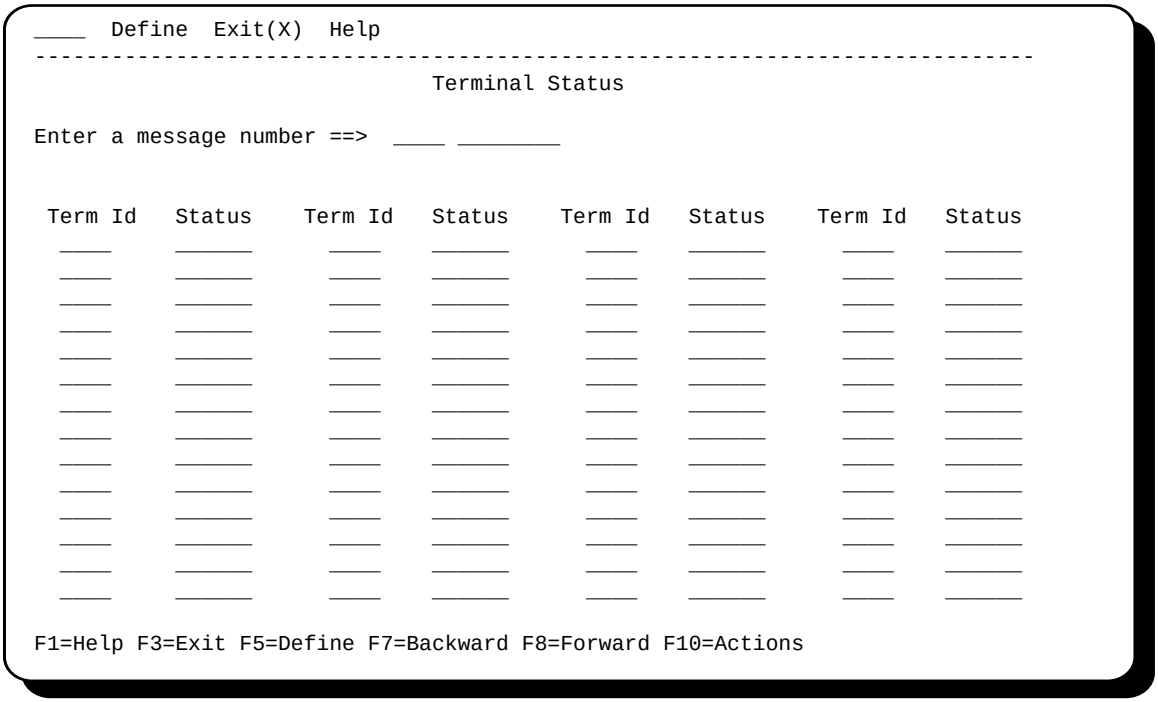
At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP	Display a Help screen for information pertaining to this screen.
PF3 EXIT	Exit to the Message Directory display.
PF5 DEFINE	This is used to view and optionally modify the message or the parameters of the message to which the cursor pointed at the time PF5 is pressed or by the name keyed in the ==> field.
PF6 STATUS	Display the Terminal Status screen for the message to which the cursor pointed at the time PF6 is pressed or by the ==> field.
PF7 BACKWARD	Browse backward to the previous page.
PF8 FORWARD	Browse forward to the next page.
PF10 ACTIONS	Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE TERMINAL STATUS DISPLAY

The Terminal Status screen displays a summary of the destinations that are specified by the message shown in the ==> field. The Terminal Status screen is displayed by pressing PF6 from the Active Message Status display. It appears as follows:


____ Define Exit(X) Help
Terminal Status
Enter a message number ==> ____ ________
Term Id Status Term Id Status Term Id Status Term Id Status
F1=Help F3=Exit F5=Define F7=Backward F8=Forward F10=Actions

FIELDS OF THE TERMINAL STATUS DISPLAY

STATUS This is the status of the terminal. Possible values to be displayed in this field are:

WAITING The terminal has not yet received the message.

IN PROGRESS The terminal is currently displaying the message.

TERM ID This is the ID of the terminal that has been targeted to receive the message. Once a terminal has received the message, the terminal will no longer appear on this list.

ENTERING COMMANDS AT THE TERMINAL STATUS DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF3 EXIT Exit to the Active Message Status display.

PF5 DEFINE This is used to view and optionally modify the message or the parameters of the message to which the cursor pointed at the time PF5 is pressed or by the name keyed in the ==> field.

PF7 BACKWARD Browse backward to the previous page of terminal entries.

PF8 FORWARD Browse forward to the next page of terminal entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PF10 ACTIONS	Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

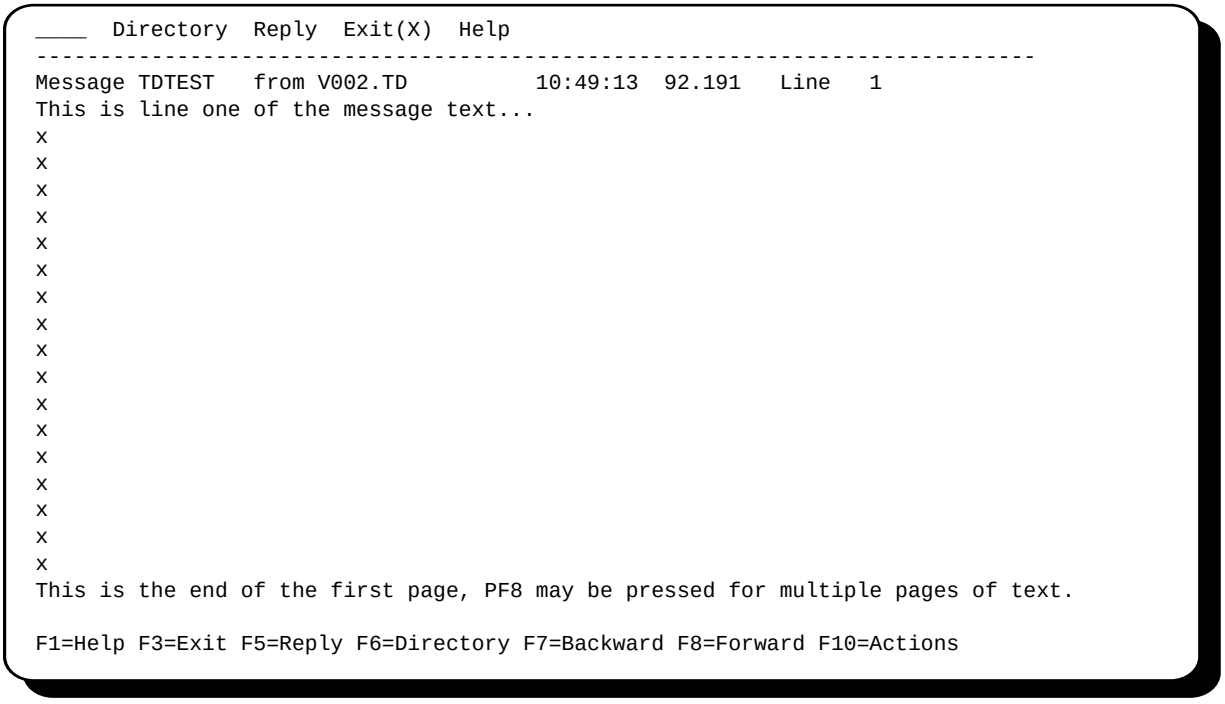
© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MESSAGE RECEPTION DISPLAY

When a message has been transmitted, either the message will be immediately received, or it will be queued and will wait until the operator requests the message to be displayed, depending on the coding of the Receive Type options of the message definition.

Once the message is displayed by either receive type, the message will be displayed in the phantom session. The Phantom Session is a temporary session for CICS-WINDOWS internal use only. This session is in addition to the number of sessions that are present on this terminal. It is made available when needed by CICS-WINDOWS and disappears once the need for this session is resolved.

The message is displayed on the Message Reception Display, which appears generally as follows:


____ Directory Reply Exit(X) Help
Message TDTEST from V002.TD 10:49:13 92.191 Line 1
This is line one of the message text...
This is the end of the first page, PF8 may be pressed for multiple pages of text.
F1=Help F3=Exit F5=Reply F6=Directory F7=Backward F8=Forward F10=Actions

Along with the message text, the Message Reception screen displays the message ID, the terminal and operator IDs of the terminal that sent the message and the time and date the message was sent.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE MESSAGE RECEPTION DISPLAY

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

PF1 HELP — Display a Help screen for information pertaining to this screen.

PF3 EXIT — This will exit the display and return to the transaction that was executing prior to receiving the message.

PF5 REPLY — This will display the message editor, in order to allow a response to be sent to the terminal that transmitted the message. Fore more information about the editor, see THE MESSAGE TEXT EDITOR, earlier in this section.

PF6 DIRECTORY — This will display the Message Directory Display, just as if the message broadcasting facility was entered through the WMSG transaction code. The only difference being, when PF3 is pressed from the Message Directory display (to exit the transaction), the current screen will be displayed instead of returning to CICS.

PF7 BACKWARD — Browse backward to the previous page of message text.

PF8 FORWARD — Browse forward to the next page of message text.

PF10 ACTIONS — Remember the cursor position, and place the cursor in the action bar. This would be used for action bar function that are cursor position dependant.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 10. INSTALLATION OF THE CICS-WINDOWS PRODUCT

This section describes the procedures for installing CICS-WINDOWS in your CICS environment. Procedures for both DOS/VSE and OS/MVS are described.

CICS REQUIREMENTS (*** MUST READ ***)

1) The parameter EXITS=YES must be specified in the System Initialization Table or as a SIT override.

2) The module DFHPCPxx must run in the same partition or region as CICS. That is, for MVS, these modules can not reside in the LPA and for DOS, they can not reside in the SVA. (xx is the CICS Module suffix).

   CICS 3.3 users must either unprotect the extended read-only DSA (RENTPGM=NOPROTECT in DFHSIT) or remove the reentrant flag from the DFHPCP load module, which will cause it to load in storage protect key 8, rather than zero.

3) If your CICS contains a Nucleus Load Table (NLT) with PROTECT=YES, the DFHPCPxx must not be included in the NLT (xx is the CICS Module suffix).

4) For VTAM systems, if the READ BUFFER keyword of the CICS-WINDOWS customization table is coded YES, the maximum TIOAL parameter of the CICS Terminal Control Table may need to be increased. If you experience ATNI abends (particularly on remote terminals), or if your terminal hangs when the Toggle key is pressed, increasing this value will usually correct the problem.

5) CICS-WINDOWS will run on CICS versions 1.7 through 3.3, inclusive.

INSTALLATION STEPS

Please refer to Appendix C of this manual for installation steps and jcl.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE INSTALLATION TAPE

The installation tape for VSE installations consists of three files.

FILE1 CSH.WNDO42.VSE.TAPE.FILE1
FILE2 CSH.WNDO42.VSE.TAPE.FILE2
FILE3 CSH.WNDO42.VSE.TAPE.FILE3

The installation tape for MVS installations consists of four files.

FILE1 CSH.WNDO42.INSTLIB
FILE2 CSH.WNDO42.CTS130.LOADLIB (CICS TS 1.3 installation)
FILE3 CSH.WNDO42.CTS220.LOADLIB (CICS TS 2.2 & 2.3 installation)
FILE4 CSH.WNDO42.CONTROL.FILE

KEYWORDS OF THE UNICOM INSTALLATION PROGRAM

The installation program, STSINST, which is provided on the first file of the installation tape can be used to print the documentation or perform a complete install for any of the products on the master installation tape.

The individual installation instructions of each product provide the necessary JCL and keyword parameters that are specific to that product.

This section describes all of the available keywords of the STSINST program, plus the coding format used.

FORMAT OF THE STSINST KEYWORDS

All keywords and operands of the STSINST program are coded in the SYSIN (MVS) or SYSIPT (DOS) dataset. Each keyword and operand occupies one statement. You cannot concatenate multiple keywords on one line. Each keyword must begin in position one.

Following are all of the possible keywords and operands of the STSINST program, listed in alphabetical order. A detailed explanation of each keyword and operand follows.

Underlined values listed are the default values if the associated operand is omitted. Values separated by a vertical bar (|) indicate mutually exclusive values which may be coded.

KEYWORD DESCRIPTIONS

DIRECTORY The DIRECTORY keyword requires no operands. If present, it will print a directory of the tape contents.

CICS=160 | 170 | 210 | 211 | 212 | 311 | 321 | 330
This is the CICS release level. Valid entries are 160, 170 and 210 for DOS/VSE. For MVS: 170, 210, 211, 212, 311, 321, and 330. This keyword provides two functions. The primary purpose is to ensure that the correct version of CICS-WINDOWS is installed for your CICS release. The second function

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

is to automatically adjust the JCL for the default dataset names. (i.e. for CICS=330, when referencing the install library, the punched JCL will reference CICS330.INSTLIB.)

CSDLIB=??????.??????

This is an optional keyword and is intended for MVS use only. This is not applicable to CICS releases prior to and including CICS 2.1.2. CSDLIB is used to specify the steplib for DFHCSDUP modules. If used, this keyword must be specified in the CREATE install step.

DFHCSD=??????.DFHCSD

This is an optional keyword and is intended for MVS use only. This is not applicable to CICS releases prior to and including CICS 2.1.2. DFHCSD is used to specify the DSN for the DFHCSD file. If used, this keyword must be specified in the CREATE install step.

FSRVFILE=xxxxxx

The FSRVFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file
for CICS-FILESERV. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

HELPSCRN=xxxxx

The HELPSCRN keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM screen file for HELP-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

HELPTEXT=xxxxxx

The HELPTEXT keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM text file for HELP-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

INSTLIB=??????.??????

This is an optional keyword and is intended for MVS use only. INSTLIB is used to specify the steplib for the STSINST module. If used, this keyword must be specified in the CREATE install step.

JCL=INSTALL|REINSTALL

The JCL keyword defines the action to be taken for the installation job control statements. Valid operands are:

INSTALL	Create JCL which will delete and redefine the VSAM file(s).
REINSTALL	Create JCL which will not delete nor redefine the VSAM file(s).

If the JCL keyword is omitted, the installation JCL will be printed only. If either INSTALL or REINSTALL is specified, the JCL will be punched to the POWER punch queue (DOS) or written to the PDS member identified by the SYSPCH DD statement (MVS).

JUGLFILE=xxxxxx

The JUGLFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-JUGGLER or CICS-JUGGLER/SVT. The xxxxxx is the 1-7 byte file ID of the DLBL statement (DOS) or 1-8 byte DDNAME of the DD statement (MVS).

LINES=56|nn

The LINES keyword defines the number of lines to print on the system printer before skipping to a new page. If omitted, 56 is the default.

MODE=PRINT|CREATE|INSTALL|REINSTALL|DOCUMENT

The MODE keyword defines the action to be taken by STSINST. Valid mode operands are:

PRINT	Print the installation instructions for the specified product.
CREATE	Create an installation file on the work tape for the specified product.
INSTALL	Load the VSAM files associated with the specified product. Coding MODE=INSTALL indicates that this is a new installation, loading to an empty VSAM file.
REINSTALL	Load the VSAM files associated with the specified product. MODE=REINSTALL indicates that this is a subsequent installation, reloading the system records to an existing VSAM file that contains user data.
DOCUMENT	Print the overview documentation for the specified product. The overview document is a short preview of the product, telling you how to get started using it. If you need the complete reference guide, that can be obtained from your sales representative.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

OPSYS=DOS|MVS|MVS/XA|MVS/SP

The OPSYS keyword defines the operating system where the specified product is to reside. For all products except VTAM-EXPRESS, you need only to specify DOS or MVS. For the MVS version of EXPRESS you must specify either MVS/XA or MVS/SP. FOR MVS/ESA, specify MVS/XA.

PRODUCT=WINDOWS|JUGGLER|SVT|HELP|FILESERV|VWINDOWS|EXPRESS

The PRODUCT keyword identifies the product to be installed, or for which documentation is to be printed. A brief description of each product is contained in Appendix D.

The full product name corresponding to each keyword is:

WINDOWS	-	CICS-WINDOWS
JUGGLER	-	CICS-JUGGLER
SVT	-	CICS-JUGGLER/SVT
HELP	-	HELP-WINDOWS
FILESERV	-	CICS-FILESERV
VWINDOWS	-	VTAM-WINDOWS
EXPRESS	-	VTAM-EXPRESS

SYSLIB=??????.??????

This is an optional keyword and is intended for MVS use only. SYSLIB is used to specify the SYSLIB for the IEWL execution. If used, this keyword must be specified in the CREATE install step.

SYSLMOD=??????.??????

This is an optional keyword and is intended for MVS use only. SYSLMOD is used to specify the SYSLMOD for the link edit of the CICS-WINDOWS programs. If used, this keyword must be specified in the CREATE install step.

SYSUT1=??????.??????

This is an optional keyword and is intended for MVS use only. SYSUT1 is used to specify the PDS to place the source and macro catalogs. If used, this keyword must be specified in the CREATE install step.

WNDOFILE=xxxxx

The WNDOFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for CICS-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

VTWOFILE=xxxxxx

The VTWOFILE keyword identifies the name of the DLBL or DD statement in the job control statements for this execution of STSINST that names the VSAM file for VTAM-WINDOWS. The xxxxxx is the file ID of the DLBL statement (DOS) or DDNAME of the DD statement (MVS).

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INSTALLING THE PRODUCT CONTROL PASSWORD

The product control password is a special code which controls the authorized use of any product from Unicom Systems. During the trial evaluation period, the product control password defines the date at which the trial version of the product will expire and can no longer be used.

For permanent licensed users, the product password defines all of the CPUs where the product may be used. Passwords for lease or lease-purchase licenses contain an expiration date and CPU ID.

When you initially receive an installation tape, whether for a trial evaluation or not, the tape contains a temporary password that will keep the product from expiring for approximately 30 days. If you are a licensed user, part of the re-install procedure is to assemble your permanent password table after installing the new tape.

If you are evaluating the product and have received a new temporary password from your sales representative, you must assemble the password table with the new password to keep the product from expiring.

The following job control is used to assemble and catalog the password control table, STSPASS.

DOS/VSE USERS

```
// JOB PASSWORD ASSEMBLY
// LIBDEF PHASE,CATALOG=L.S                [Note 1]
// OPTION CATAL
   PHASE STSPASS,*      [Note 2]
// EXEC ASSEMBLY
      DC    CL6'nnnnnn'                   CPU ID
      DC    CL4'xxxx'                     Product ID
      DC    CL6'mmddyy'                   Expiration Date
      DC    CL16'xxxxxxxxxxxxxxx'         Password
      .
      .
      .
      DC    CL6'nnnnnn'                   CPU ID
      DC    CL4'xxxx'                     Product ID
      DC    CL6'mmddyy'                   Expiration Date
      DC    CL16'xxxxxxxxxxxxxxx'         Password
      END
/*
// EXEC LNKEDT
/&
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MVS USERS

```
//PASSTBL       JOB     1.'ACCOUNT DATA',MSGCLASS=x
//ASM           EXEC    PGM=IEV90,PARM='DECK'
//SYSPRINT      DD      SYSOUT=*
//SYSLIN        DD      DUMMY
//SYSUT1        DD      UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH DD   UNIT=SYSDA,DISP=(,PASS),
//                      SPACE=(CYL,(1,1)),
//                      DSN=&&TEMP1,
//                      DCB=BLKSIZE=3200
//SYSIN         DD      *
                DC      CL6'nnnnnn'                     CPU ID
                DC      CL4'xxxx'                       Product ID
                DC      CL6'mmddyy'                     Expiration Date
                DC      CL16'xxxxxxxxxxxxx   '          Password
                .
                .
                .
                DC      CL6'nnnnnn'                     CPU ID
                DC      CL4'xxxx'                       Product ID
                DC      CL6'mmddyy'                     Expiration Date
                DC      CL16'xxxxxxxxxxxxx   '          Password
                END
                /*
//LINK          EXEC    PGM=IEWL,PARM=(LIST,XREF),
//                      COND=(4,LT)
//SYSLIB        DD      DUMMY
//SYSPRINT      DD      SYSOUT=*
//SYSLIN        DD      DSN=&&TEMP1,
//                      DISP=(OLD,DELETE)
//SYSLMOD       DD      DSN=CICS.STEPLIB(STSPASS),      [Note 1]
//                      DISP=SHR                        [Note 2]
//SYSUT1        DD      UNIT=SYSDA,SPACE=(CYL,(1,1))
```

[Notes]:

1. The library containing the link-edited module must be a CICS load library.
2. The link-edited module name must be STSPASS

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DATA FIELDS OF THE PASSWORD CONTROL TABLE

The password table is a series of define constant instructions. The same password table is used for all products from Unicom Systems.

There are four DC statements required for each product, or for each occurrence of the same product on multiple CPUs. There is no limit to the number of product entries that may be present. The four DC instructions define the following values:

1). The first entry in the table is a 6-position CPU ID. For temporary passwords, code this as all zeros. For licensed users, code the full 6-position CPU ID where the product will operate.

If you are a VM user and have multiple guest machines on the same CPU, you can get by with a single entry if the last five positions of the CPU ID on each guest machine is the same. If this is the case, code zero as the first digit, followed by the five identifying digits. If each guest CPU ID is different, you must make an entry for each unique CPU ID.

2). The second entry is a 4-position product ID. Codes are:

WNDO - CICS-WINDOWS
JUGL - CICS-JUGGLER
SVT - CICS-JUGGLER/SVT
HELP - HELP-WINDOWS
FSRV - CICS-FILESERV
VTWO - VTAM-WINDOWS
EXPR - VTAM-EXPRESS

3). The third entry is the expiration date. For temporary passwords or for lease and lease-purchase licenses, you will be given an expiration date along with your password. That date must be coded here in MMDDYY format. For perpetual licenses, code this as all zeros.

4). The last entry is the password itself. You must define it as a 16-byte field, although the total number of characters in the password may not reach 16. Code the password that you have been given.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 11. CUSTOMIZATION

Certain default options in CICS-WINDOWS may be modified by the CICS system programmer. This tailoring could be performed for security reasons or for ease of operation. The following list provides most of the options along with the name of the table with which they are associated:

1) Limit the maximum number of CICS-WINDOWS users at one time. (User Option table)

2) Limit the maximum virtual terminals per physical terminal to less than nine. (User Option Table)

3) Exclude certain terminals or users from being able to use CICS-WINDOWS. (Terminal Exclusion Table)

4) Exclude certain transactions from recognizing the "Hot" keys. (Transaction Exclusion Table)

5) Force CICS-WINDOWS termination at CICS sign-on and sign-off. (User Option Table)

6) Eliminate the Pseudo Terminal ID option. (User Option Table)

7) Control the specification of Pseudo Terminal IDs. (User Option Table, Auto-Init Table, PSGNXIT statement in the WNDOTBL macro)

8) Define certain programs or transactions as "single-occurring", meaning that they can only be active in one virtual terminal per physical terminal at a time. (Single Occurring Transactions Table, Single Occurring Programs Table)

9) Prevent operators from terminating CICS-WINDOWS without first terminating the transactions in all virtual terminals. (User Option Table)

10) Define transaction codes other than CSSF and CSSN as sign-off or sign-on transactions. (Signoff Transactions Table, Signon Transactions Table)

11) Establish security for certain WNDO commands. (User Option Table)

12) Pre-define any or all start-up options of CICS-WINDOWS for each terminal. (Profile Table)

13) Define additional function keys to be used by terminal operators. (Profile Table) Some of which are:

- Direct-session keys; the ability to assign a PF or PA key to each virtual terminal, thereby allowing direct access to each session with one keystroke.
- Toggle Backward key, allows toggling in reverse sequence.
- Window switch keys, which allow the assignment of a PF or PA key to each window, which, when pressed, will effect a Window Switch command.
- Scrolling keys, allowing the assignment of PF or PA keys to the four scrolling functions, Up, Down, Right and Left.

14) Provide support for multiple Interactive Interface sessions for DOS/VSE users. (Profile Table)

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

15) Suppress the terminal READ BUFFER command. (User Options Table, Profile Table)

16) Suppress the User Configuration display at WNDO,ON time. (User Options Table)

17) Maintain the same terminal user area in all virtual terminals. (User Options Table)

18) Define the window mode configuration default. (User Options Table, Profile Table)

19) Enable the data stream compression feature. (User Options Table)

20) Exclude certain transactions from participating in data compression. (Transaction Compression Table)

21) Exclude certain terminals from participating in data compression. (Terminal Compression Table)

22) Specify a time interval, after which all or selected conversational tasks which have had no activity will be automatically purged. (User Options Table)

23) Define a user transaction to be automatically started at a terminal following CICS-WINDOWS activation and when first toggling into an empty session. (Application Startup Table)

24) Define a table of transactions which can not be run if CICS-WINDOWS is active on the terminal. (Stopped Transactions Table)

25) Exclude certain transactions from running in a window. (Window Mode Stopped Transactions)

26) Designate a message log Transient Data queue for CICS-WINDOWS to use. (User Options Table)

27) Disable the Temporary Storage key modification feature. (User Options Table)

28) Define a user exit program for generating Pseudo Terminal IDs. (PSGNXIT statement in the WNDOTBL macro)

29) Define a user exit program to get control when CICS-WINDOWS attaches a transaction in window mode. (ATCHXIT statement in the WNDOTBL macro)

30) Associate a different VSAM file name with an alternate transaction code. (File Table)

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CONVERTING THE WNDOTBL MACRO TO THE ONLINE VERSION

In past releases, all customization options were specified by the use of a User Option Table which had to be coded by means of the WNDOTBL macro statements, assembled, link-edited and included in CICS by means of a PPT entry. Now, nearly all of the customization options can be performed interactively, on-line. The WNDOTBL macro is still provided with the package and must be used for three options, which are the name of the file where the on-line customization tables are to be kept and to enable the use of two user exits, explained later in this section.

Previously assembled customization tables may be converted to the on-line format simply by entering the WAUX Transaction (explained later in this section, under *THE AUXILIARY FUNCTIONS TRANSACTION*), which if no dynamic options are present, will display a screen asking if you wish to migrate the old WNDOTBL macro program to the newer dynamic version. If you answer YES, the table will be migrated and the Auxiliary Functions Transaction menu will be displayed. Subsequent invocations of this transaction will display the menu without asking to migrate. Once migration has been performed, you can remove the PPT table entry for WNDOTBL. If you choose not to migrate the table, the program will return to CICS.

CICS-WINDOWS will operate without converting the old WNDOTBL macro to the newer dynamic version; however, we strongly recommend conversion. The dynamic process is much more friendly, and nearly all changes made to the dynamic table will take effect immediately without the need to interrupt any operators that may currently be operating under CICS-WINDOWS. In addition, conversion from much older releases may not always be supported.

STEPS OF CONVERTING THE WNDOTBL MACRO

If migrating the WNDOTBL macro to the dynamic version is desired at this time, follow these steps:

1) Ensure that CICS-WINDOWS is not initialized. This may be verified by using CEMT to see if program WINDOWS is resident. If it is resident, you must issue the WNDO,BACKOUT command.
2) NEWCOPY all CICS-WINDOWS programs.
3) Ensure that the WNDOTBL program does have a PPT entry and it is enabled.
4) Ensure that no "O" prefix records exist in the WNDOFIL. This may be verified with the command **CECI READ DATASET(WNDOFIL) RIDFLD(O) GENERIC**. There must **not** be any option records, as this is one check that is performed by CICS-WINDOWS in the migration process. The WNDOFIL is distributed without any option records.
5) ENTER the **WAUX** transaction code. At this time, a screen should display asking if you wish to migrate the WNDOTBL macro.
6) If you selected the migrate option, the records will be converted and written to the WNDOFIL.

If the migration was successful, you may remove the WNDOTBL program from the PPT. If for some reason the migration was not successful, or if after following the above instructions, you were not prompted for migration, contact Unicom Systems technical support as explained in the appendix.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDOTBL MACRO STATEMENT FORMAT

Following are all the possible operands of the WNDOTBL macro, along with an explanation of each. This table must be coded if any of the three operands are required for your environment, otherwise it is not necessary. As explained above, nearly all options of the WNDOTBL macro are now controlled through on-line customization, so the remaining customization options are explained in ON-LINE CUSTOMIZATION, later in this section.

Underlined values listed are the default values if the associated operand is omitted. Values separated with a vertical bar (|) indicate mutually exclusive values that may be coded.

```
WNDOTBL     ATCHXIT=[NO | symbol],                    X
            OPTFILE=(WNDOFIL | symbol),               X
            PSGNXIT=[NO | symbol],
```

ATCHXIT This is used to specify the name of a program that is to get control any time CICS-WINDOWS is ready to initiate a transaction in window mode. For instance, this exit program can alter the transaction code to some other transaction based on your security requirements. If this feature is desired, code the name of your program, else code "NO" or omit.

For more information, see *WRITING A USER EXIT FOR CICS-WINDOWS* in section 15 - *USER EXITS AND PROGRAM INTERFACES*.

OPTFILE This is used to specify the name of the file to be used by CICS-WINDOWS for saving customization options and other internal control records. The default is "WNDOFIL". If you wish to use a different file name, you must code the name of the file that was created during the installation process.

PSGNXIT This is used to specify the name of a user exit program that will get control during a WNDO,ON or WNDO,INIT command when CICS-WINDOWS is ready to generate the Pseudo Terminal IDs. This exit program can generate a unique terminal ID for each virtual terminal using any method desired. If this feature is desired, code the name of your program, else code "NO" or omit.

For more information, see *WRITING A USER EXIT FOR CICS-WINDOWS* in section 15 - *USER EXITS AND PROGRAM INTERFACES*.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INSTALLATION OF THE WNDOTBL MACRO

The following JCL examples can be used to assemble and link-edit the WNDOTBL macro table for CICS-WINDOWS.

DOS Assembly and Link-Edit

```
// JOB WNDOTBL ASSEMBLY
// LIBDEF CL,TO=CICSCL                                    [Note 3]
// OPTION CATAL
  PHASE  WNDOTBL,*
// EXEC ASSEMBLY
        WNDOTBL ATCHXIT=xxxxxxxx,                              X
                     OPTFILE=xxxxxxxx,                         X
                     PSGNXIT=xxxxxxxx
        END
/*
// EXEC LNKEDT
/&
```

MVS Assembly and Link-Edit

```
//WNDOTBL       JOB                    1.'ACCOUNT-DATA',MSGCLASS=X
//ASM   EXEC    PGM=IEV90,PARM='DECK'
//SYSLIB        DD                     DSN=CICS.MACLIB,DISP=SHR  [Note 1]
//        DD    DSN=SYS1.MACLIB,DISP=SHR          [Note 2]
//SYSPRINT      DD                     SYSOUT=*
//SYSLIN        DD                     DUMMY
//SYSUT1        DD                     UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH      DD                     UNIT=SYSDA,DISP=(,PASS),
//                    SPACE=(CYL,(1,1)),
//                    DSN=&&TEMP1,
//                    DCB=BLKSIZE=3200
//SYSIN DD      *
          WNDOTBL ATCHXIT=xxxxxxxx,                            X
                       OPTFILE=xxxxxxxx,                       X
                       PSGNXIT=xxxxxxxx
          END
/*
//LINK  EXEC    PGM=IEWL,PARM=(LIST,XREF),
//                    COND=(4,LT)
//SYSLIB        DD                     DUMMY
//SYSPRINT      DD                     SYSOUT=*
//SYSLIN        DD                     DSN=&&TEMP1,DISP=(OLD,DELETE)
//SYSLMOD       DD                     DSN=CICS.LOADLIB(WNDOTBL),     [Note 3]
//                    DISP=SHR
//SYSUT1        DD                     UNIT=SYSDA,SPACE=(CYL,(1,1))
```

[Notes]:

1. Macro library for CICS (where WNDOTBL macro was loaded)
2. MVS System macro library
3. Core-Image / Load library for CICS applications

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ON-LINE CUSTOMIZATION

There are fifteen categories of customization options that can be performed on-line These are:

1) Define the options of the "User Option" table. Most of the global options that were previously supplied by the WNDOTBL macro are now defined here.

2) Define a File table. This table is used when multiple CICS-WINDOWS VSAM files are to be used in one CICS system. This might be desirable when one or more different transaction codes (other that WNDO) are to be used, so that all users of a particular transaction code will use the same VSAM file, while other users will access a different file.

3) Customize "User Profiles". This table is used to pre-define some or all of the start-up options for a given terminal, or for all terminals. This enables operators to have pre-defined CICS-WINDOWS options that differ from or augment the Global User Option settings.

4) Define an "Auto-Init" table. This table is used by CICS-WINDOWS to associate an operator with a User Profile when a WNDO,ON command has been issued by that operator. In addition, this table can also be used to assign preselected pseudo terminal IDs to that operator.

5) Define an "Application Startup" table. This table provides a means of starting a user transaction automatically in each virtual terminal. You may define the same or different transactions in each session, and you may define data to be passed to the application program to vary the operation in each session.

6) Define a "Terminal Exclusion" table. This table is used to define one or more terminal IDs or operators that are to be excluded from using CICS-WINDOWS or included as the only terminals that can use CICS-WINDOWS.

7) Define a "Transaction Exclusion" table. This table is used to define one or more transaction codes that are to be excluded from recognizing the hot keys or included as the only transactions that will recognize the hot keys.

8) Define a "Sign-off Transaction" table. This table is used to define one or more transaction codes that designate a sign-off transaction, if you are using a transaction codes other than CSSF or CESF.

9) Define a "Sign-on Transaction" table. This table is used to define one or more transaction codes that designate a sign-on transaction, if you are using transaction codes other than CSSN OR CESN.

10) Define a "Terminal Compression" table. This table is used to define a table of terminal IDs that either are not to participate in the data compression feature of CICS-WINDOWS, even though the data compression feature is on, or are to be the only terminals that are eligible to participate in the data compression.

11) Define a "Transaction Compression" table. This table is used to define a table of transaction codes that are either not to participate in the data compression feature of CICS-WINDOWS, even though the data compression feature is on, or are to be the only transactions that will be eligible to participate in compression.

12) Define a "Single Occurring Transaction" table. This table is used to define one or more transaction codes that are "single-occurring", meaning it can only be active in one virtual terminal per physical terminal at a time.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

13) Define a "Single Occurring Program" table. This table is used to define one or more program names which are "single-occurring".

14) Define a "Stopped Transactions" table. This table is used to define one or more transaction codes that will not be allowed to execute if CICS-WINDOWS is active on the terminal.

15) Define a "Window Mode Stopped Transactions" table. This table is used to define one or more transaction codes that will not be allowed to execute in window mode.

The relationships of the various on-line customization tables are as follows:

1) The User Options table specifies the global defaults.

2) The User Profile overrides or augments the User Option table for one or more operators. It may also be used to link the 'owner(s)' of the profile to a Start Transaction table.

4) The Auto-Init table links operators and terminals to their respective User Profiles.

5) The Sign-on and Sign-off tables are used to define sign-on and sign-off transactions to CICS-WINDOWS, which may use the information when working with certain customization options that deal with signing-on and/or signing-off.

6) The remaining tables have specific functions that do not directly hinge on other tables.

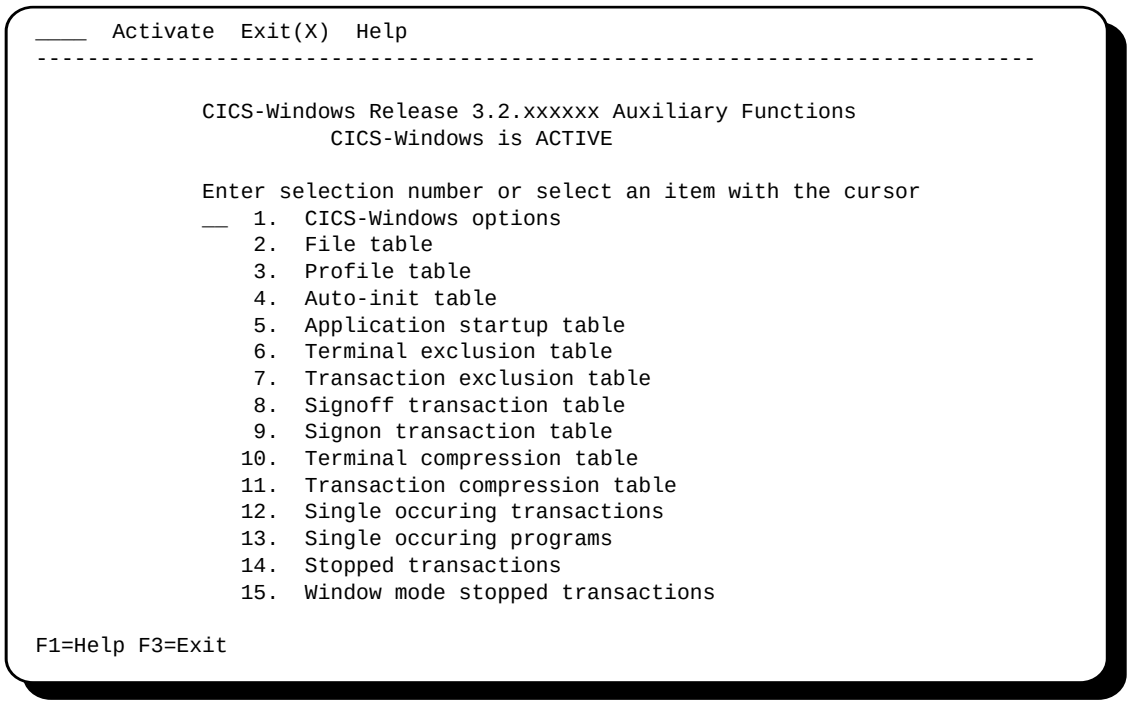
The dynamic customization options are accessed through the use of the Auxiliary Functions transaction - WAUX, which is explained in more detail in the following discussion.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE AUXILIARY FUNCTIONS TRANSACTION

The Auxiliary Functions transaction is used for specifying nearly all the customization options for CICS-WINDOWS.

To invoke the Auxiliary Functions transaction, key the transaction code WAUX from a clear screen. Upon pressing ENTER the Auxiliary Functions menu will appear:


____ Activate Exit(X) Help
CICS-Windows Release 3.2.xxxxxx Auxiliary Functions
CICS-Windows is ACTIVE
Enter selection number or select an item with the cursor
__ 1. CICS-Windows options
2. File table
3. Profile table
4. Auto-init table
5. Application startup table
6. Terminal exclusion table
7. Transaction exclusion table
8. Signoff transaction table
9. Signon transaction table
10. Terminal compression table
11. Transaction compression table
12. Single occuring transactions
13. Single occuring programs
14. Stopped transactions
15. Window mode stopped transactions
F1=Help F3=Exit

[Note]: If the dynamic user options table record is not found in the WNDOFIL and a WNDOTBL program is found in this CICS, a different screen will display asking if you wish to migrate the old WNDOTBL macro to the newer dynamic customization version. If you choose to migrate, the WNDOTBL macro will be converted and then the above screen will display.

ENTERING COMMANDS AT THE AUXILIARY FUNCTIONS MENU

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER — View and optionally modify the table that was specified in the entry field, or the table to which the cursor was pointing when ENTER was pressed

PF1 HELP — Display a Help window for information pertaining to the field that the cursor is pointing when PF1 is pressed.

PF3 EXIT — Exit the Auxiliary Functions transaction.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ACTION BAR FUNCTIONS AT THE AUXILIARY FUNCTIONS MENU

The following actions are available from the action bar:

ACTIVATE **Start** CICS-WINDOWS This has the same effect as issuing a WNDO,START command. This is used to cancel a previous STOP command and allows operators to perform a WNDO,ON command.

Stop CICS-WINDOWS This has the same effect as issuing a WNDO,STOP command. This will prevent operators from completing a WNDO,ON command until a START command is issued

Backout CICS-WINDOWS This has the same effect as issuing a WNDO,BACKOUT command. This is used for completely deactivating WINDOWS, usually to allow the application of a PTF or to allow WINDOWS to be newcopied.

For more information on START, STOP and BACKOUT, please refer to section 14 - *SPECIAL-PURPOSE COMMANDS*.

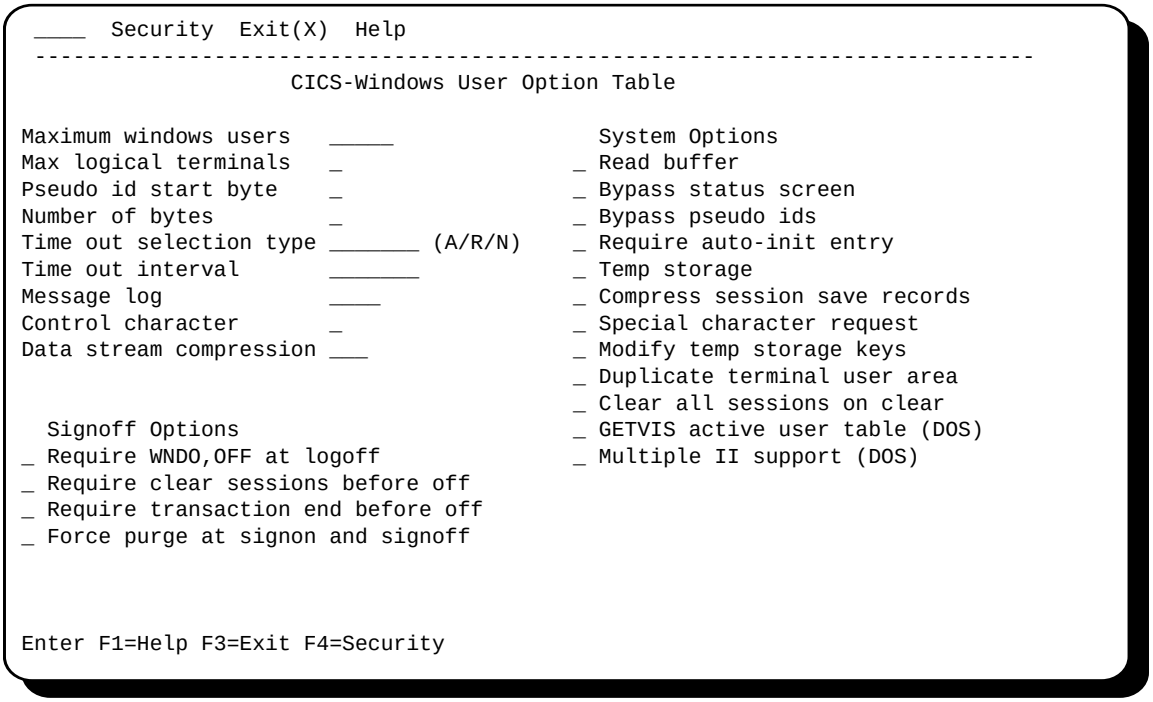
EXIT End Transaction This has the same effect as PF3, which simply exits to CICS.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE USER OPTIONS TABLE

The User Options table is used for setting most of the Global User Options available for CICS-WINDOWS.

To display the User Options table, from the Auxiliary Functions menu key a "1" and press ENTER. This screen will appear generally as follows:


____ Security Exit(X) Help
CICS-Windows User Option Table
Maximum windows users _____
Max logical terminals _
Pseudo id start byte _
Number of bytes _
Time out selection type _______ (A/R/N)
Time out interval _______
Message log ____
Control character _
Data stream compression ___
System Options
_ Read buffer
_ Bypass status screen
_ Bypass pseudo ids
_ Require auto-init entry
_ Temp storage
_ Compress session save records
_ Special character request
_ Modify temp storage keys
_ Duplicate terminal user area
_ Clear all sessions on clear
_ GETVIS active user table (DOS)
_ Multiple II support (DOS)
Signoff Options
_ Require WNDO,OFF at logoff
_ Require clear sessions before off
_ Require transaction end before off
_ Force purge at signon and signoff
Enter F1=Help F3=Exit F4=Security

You can change any value where the cursor will stop when the TAB key is pressed (except the PF key line). The following discussion describes each field.

FIELDS OF THE ON-LINE USER OPTION TABLE

BYPASS PSEUDO IDS

This field is used to specify whether pseudo terminal IDs are to be generated. It applies to any terminal using CICS-WINDOWS, which does not have pseudo terminal IDs defined in an Auto-Init table entry, either specifically or generically. It also applies if CICS-WINDOWS is activated with the WNDO,INIT command instead of the WNDO,ON command. Values are:

ON No additional terminal IDs are needed. CICS-WINDOWS will not allow operators to enter pseudo terminal IDs upon start-up.

OFF CICS-WINDOWS will generate pseudo terminal IDs as specified in the PSEUDO ID START, NUMBER OF BYTES and SPECIAL CHARACTER REQUEST fields or when ready to generate pseudo IDs.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

BYPASS STATUS SCREEN

This option is used to specify whether the User Configuration screen, which is normally produced when a WNDO,ON, WNDO,INIT or WNDO,OFF command is issued, is to be bypassed when using the WNDO,ON or WNDO,OFF commands.

One purpose of bypassing the User Configuration Display would be when activating and terminating CICS-WINDOWS under program control (see *INITIALIZING CICS-WINDOWS FROM A USER PROGRAM*). It may be that the user program produces a display after linking to the WINDOWS program. Without this option, the User Configuration Display would appear on the terminal as an intermittent flash prior to the user program display, which is usually not desired.

This option will not affect whether the User Configuration screen is displayed in response to the WNDO, WNDO,INIT or WNDO,INQ commands.

This is an ON/OFF type option:

ON The User Configuration screen is to be bypassed.

OFF The User Configuration screen is not to be bypassed.

COMPRESS SESSION SAVE RECORDS

This is used to specify whether the record held in either Temporary Storage or MAIN storage, for each terminal session, will be compressed by removing all repetitive characters. This can result in significant savings of file and memory space since these records contain an image of the screen for each session.

ON Compress the session save records.

OFF Do not perform any compression algorithms on the session save records.

CONTROL CHARACTER

This is used to specify a character to be used with control character commands such as Cut and Paste.

DATA STREAM COMPRESSION

This field is used to specify whether datastreams compression is to be active or not. Values for this field are:

NO Do not perform any datastream optimization.

YES Optimize datastreams only for terminals that are using CICS-WINDOWS.

ALL Optimize datastreams for all terminals in this CICS, even if the terminal is not using CICS-WINDOWS.

Data compression in CICS-WINDOWS consists of the elimination of consecutive repetitive characters on outbound datastreams, which can usually save around 30 to 40 percent of terminal transmission time. This can be quite significant, especially when remote terminals are in use.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DUPLICATE TERMINAL USER AREA

This is used to specify whether the Terminal User Area will be common to all sessions of a terminal using CICS-WINDOWS.

It is often the case that an application system will store information in the Terminal User Area (user security information, etc.) which must be the same in all sessions. Without this option, the user area is initially set for all sessions to whatever it contains at WNDO,ON time, then maintained from session to session. If an application in one session changes the value, the original value is kept in all other sessions.

With this option, the Terminal User Area is not restored when toggling into a session. Thus, what-ever value was placed in it by the previous application is carried into the next session. The only way to clear or reset the Terminal User Area is to WNDO,OFF and back on. This is an ON/OFF type option:

ON CICS-WINDOWS will maintain a common Terminal User Area in all sessions of a terminal.

OFF CICS-WINDOWS will not make any attempt to maintain a common Terminal User Area.

FORCE PURGE AT SIGNON AND SIGNOFF

This is used to specify whether CICS-WINDOWS is automatically terminated with a PURGE command before a new sign-on is accepted and before a sign-off is performed. This is an ON/OFF type option:

ON CICS-WINDOWS will be automatically terminated with a PURGE command before a new sign-on is accepted, and before a sign-off is allowed to complete. This means that any conversational transactions that are active on the terminal are abnormally ended with an AKCS abend.

OFF CICS-WINDOWS will not be automatically purged before a new sign-on is accepted or sign-off is performed.

GETVIS ACTIVE USER TABLE (DOS)

This option applies to VSE only. The Active User table is a dynamic table with an entry for each physical terminal for which CICS-WINDOWS is active. It requires approximately 256 bytes of storage per entry. This option specifies whether the table entries are to be contained in the GETVIS area or not. This is an ON/OFF type option:

ON Each entry will reside in the GETVIS area CICS partition. If there is insufficient GETVIS storage, all entries beyond that point will use DSA storage.

OFF All entries will reside in DSA storage.

MAX LOGICAL TERMINALS

This is used to establish the maximum number of virtual terminals that can be specified on any one physical terminal. This value will override a larger specification on the User Profile Table. Values are:

number A number from 2 to 9, inclusive.

MAXIMUM WINDOWS USERS

This is used to establish the maximum number of terminals in this CICS that can use CICS-WINDOWS at any one time. Values for this field are:

number A number from 1 to 34463, inclusive.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MESSAGE LOG

This is used to designate a 4-character Transient Data queue ID where you wish to route an activity log of messages from CICS-WINDOWS. Coding a destination ID establishes that you want a message to be sent to this destination every time a WNDO command is issued by a terminal operator.

If a destination ID is specified, a record will be written to the destination queue each time one of the following commands is issued from any terminal:

BACKOUT, CPROFF, CPRON, DEBUG, INIT, MAIN, OFF, ON, PURGE, START, STOP, TEMP

The output message will take the following form:

WNDO1419. CICS-WINDOWS xxxxxx, TERMINAL yyyy, USER zzz, TIME hh:mm:ss

In the message, xxxxxx is the command (ON, OFF, PURGE, etc.), yyyy is the terminal ID where the command was given, zzz is the Operator User ID at that terminal and hh:mm:ss is the current time of day.

The Transient Data queue name may be any valid destination ID in the Destination Control Table (DCT). This includes the normal CICS statistics and terminal message queues, CSMT and CSTL.

[Note]: If the specified destination ID is invalid, that is, it is not defined in the DCT, CICS-WINDOWS does not issue any warning message, the MESSAGE LOG specification is simply ignored. Values are:

symbol This is the 4 digit Transient Data queue ID where you wish to route activity log messages from CICS-WINDOWS.

NO No activity log will be produced.

MODIFY TEMP STORAGE KEYS

This is used to specify whether Temporary Storage keys containing terminal IDs are to be modified by the session number prior to reading or writing a Temporary Storage record. This exit is provided to ensure that Temporary Storage records created and retrieved by application programs are not duplicated when the same transaction is run in multiple virtual terminals on one physical terminal. Is pseudo terminal IDs are in use, this option should not be used. This is an ON/OFF type option:

ON Modify the key with the session number before reading or writing to the Temporary Storage record. By selecting this option, it will be possible in most cases to eliminate the use of pseudo terminal IDs and use the same terminal ID in all sessions.

OFF Do not modify the Temporary Storage key. If you will be using pseudo terminal IDs all of the time, the Temporary Storage Modification exit is unnecessary and therefore should not be selected. This will eliminate the overhead of an unnecessary global exit.

For more information on Temporary Storage keys, please refer to the discussion on *PSEUDO TERMINAL ID CONSIDERATIONS* in the section entitled *SPECIAL CONSIDERATIONS*.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MULTIPLE II SUPPORT (DOS)

This option applies to VSE only. This parameter sets the global support level for the multiple interactive interface feature. This is an ON/OFF type option:

ON If selected, the specification of the VSE. Int. Interface of the User Profile table will be honored.

OFF If not selected, there can be only one session per physical terminal containing the interactive interface.

NUMBER OF BYTES

This specifies the number of bytes of the pseudo terminal ID to be duplicated from the real terminal ID when CICS-WINDOWS is generating pseudo IDs. values are:

number A number from 1 to 3, inclusive.

[Note]: The sum of PSEUDO ID START and NUMBER OF BYTES can not exceed 4.

PSEUDO ID START BYTE

This specifies the starting position in each pseudo terminal ID to be duplicated from the real terminal ID when CICS-WINDOWS is generating pseudo IDs. Values are:

number A number from 1 to 4, inclusive.

[Note]: The sum of PSEUDO ID START and NUMBER OF BYTES can not exceed 4.

READ BUFFER

This is used to specify whether CICS-WINDOWS performs a terminal Read Buffer command when you press the toggle key or the window key to enter window mode. Certain non-IBM terminals will sometimes not support the Read Buffer command, causing ATNI abends when the toggle key is pressed. This is the case if you are using VTAM PASSTHRU for your terminal. This is an ON/OFF type option:

ON CICS-WINDOWS will issue a Read Buffer command.

OFF CICS-WINDOWS will not issue a Read Buffer command. You need to operate with READ BUFFER=NO if any of the following conditions is true:

1). You want to use the VIEW function of the SYS display and always see the latest output screen regardless of when the operator last toggled.

2). You wish to improve the response time for remote terminals when a toggle occurs.

[Note]: This option is used to globally enable or disable READ BUFFER. If selected, READ BUFFER can be overridden in the User Profile. If not selected here, the READ BUFFER option in the User Profile is ignored (READ BUFFER will not occur).

REQUIRE AUTO-INIT ENTRY

This is used to specify whether you wish to exclude any operators that are not specifically identified in the Auto-Init table from performing a WNDO,ON. This is an ON/OFF type option:

ON Only those operators specified in the Auto-Init table will be allowed to perform a WNDO,ON command.

OFF All operators may perform a WNDO,ON.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

REQUIRE CLEAR SESSIONS BEFORE OFF

This is used to specify whether users will need to end all tasks and clear all sessions before they are allowed to terminate CICS-WINDOWS. This is an ON/OFF type option:

ON Users will be required to end all tasks and clear all sessions, except the session from which the OFF command is entered

OFF Users will not be required to end all tasks and clear all sessions in order to terminate CICS-WINDOWS.

REQUIRE TRANSACTION END BEFORE OFF

This is used to specify whether operators will be required to end all transactions in each session before performing a WNDO,OFF command. This is an ON/OFF type option:

ON The operator must end all transactions, but is not necessary to clear the screens, in order to execute a WNDO,OFF command.

OFF The operator can issue a WNDO,OFF command to terminate CICS-WINDOWS while interactive tasks are in progress.

[Note]: Regardless of the coding of this option, conversational tasks must be terminated in order to perform a WNDO,OFF.

REQUIRE WNDO,OFF AT LOGOF

This is used to specify whether the operator will be required to deactivate CICS-WINDOWS on their terminal before they are allowed to logoff from CICS, using the logoff transactions coded in the "Sign-on/Sign-off Transaction" table. If FORCE PURGE AT SIGNON/SIGNOFF is selected, this option is ignored. This is an ON/OFF type option:

ON WINDOWS must be terminated to perform logoff.

OFF Upon logoff, the user will be prompted with an option to terminate WINDOWS.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SECURED COMMANDS

This portion of the User Option Table is accessed by pressing PF4 from the User Option Table.

These fields are used to specify whether the associated WNDO command is to be a secured WNDO function. The commands that may be secured are:

BACKOUT, CPROFF, CPRON, DEBUG, INIT, MAIN, PURGE, START, STOP, SYS, TEMP, WIN=

If any of the above commands are selected, you must define an additional transaction code in the Program Control Table for CICS, as follows:

```
DFHPCT  TYPE=ENTRY,TRANSID=WNSC,PROGRAM=WINDOWS,          X
            TWASIZE=0,TRANSEC=xx
```

Use a different security key for the WNSC transaction than for the WNDO transaction, one that only systems or other MIS personnel are authorized to use.

Now, if any of the listed commands are selected and issued with the WNDO transaction code, the following message will display:

WNDO1417. USER IS NOT AUTHORIZED FOR THIS COMMAND

The command will only be accepted if issued with the WNSC transaction code. (i.e. WNSC,STOP) These fields are ON/OFF type options:

ON The associated command is to be secured.

OFF The associated command is not to be secured.

SPECIAL CHARACTER REQUEST

This is used to specify whether special characters may be used while generating pseudo terminal IDs. For more information, please refer to the PSEUDO IDs operand of the Auto-Init table, later in this section. This is an ON/OFF type option:

ON Use special characters when generating pseudo terminal IDs.

OFF Do not use special characters when generating pseudo terminal IDs.

TEMP STORAGE

This is used to specify whether Temporary Storage is to be used instead of MAIN storage above the 16M line for saving terminal data. This is an ON/OFF type option:

ON Use Temporary Storage instead of MAIN storage.

OFF Do not use Temporary Storage. Instead, use MAIN storage above the 16M line.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TIME OUT INTERVAL

This specifies the time interval for the conversational transaction time out. This field is only valid if TIME OUT SELECTION is not coded "ALL". Entries are:

number A 1 to 6 digit number in 'hhmmss' format. That is, the first 2 digits specify from 00 to 24 hours, the second two digits specify from 00 to 60 minutes, and the last 2 digits specify from 00 to 60 seconds. If less then 6 digits is coded, the number is filled with high-order zeros.

YES This specifies that the interval will be the interval that is specified in the PCT. Coding YES is only valid if TIME OUT SELECTION TYPE=RTIMOUT.

When a timeout purge of a transaction occurs, the transaction is abnormally terminated with an AKCS abend code and the message

WNDO1515. TASK xxxx PURGED BY WNDO TIMEOUT

replaces the saved screen display in that session.

TIME OUT SELECTION TYPE

This parameter provides a means of automatically terminating all or certain conversational transactions when they have been inactive for a specified time. This parameter provides an alternative to the RTIMOUT operand of the Program Control Table, which will not work on a transaction that has been toggled out of. Values are:

ALL After toggling out of any conversational transaction, if the operator does not toggle back into that application for the specified interval, the transaction will be purged.

RTIMOUT

After toggling out of any conversational transaction which is specified in the PCT with RTIMOUT, if the operator does not toggle back into that application for the specified interval, the transaction will be purged.

NO No automatic purge of conversational tasks will take place.

[Note]: To use the TIMEOUT feature, the following entry must be present in the Program Control Table:

DFHPCT TYPE=ENTRY,TRANSID=WTMO,PROGRAM=WINDOWS

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ENTERING COMMANDS AT THE USER OPTIONS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

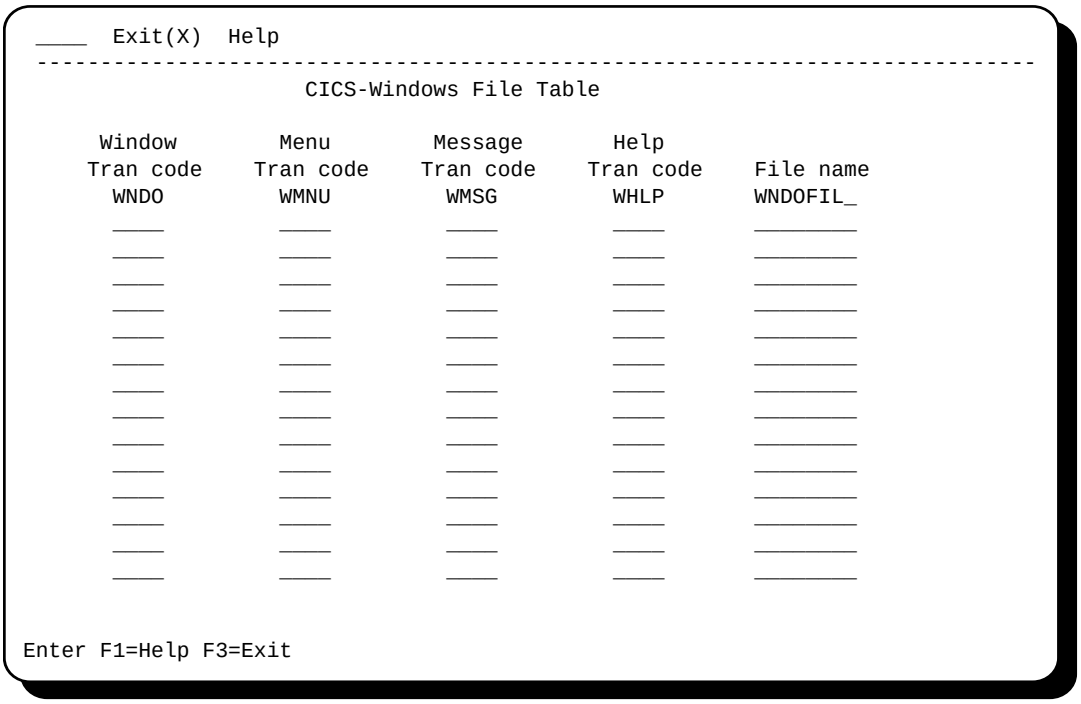
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help window for information pertaining to the field to which the cursor pointed when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF4 SECURITY	Display the Secured Commands entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE FILE TABLE

The File table is used when multiple VSAM files are to be used within one CICS system. This might be desirable when one or more different transaction codes are to be used. This statement provides a means of associating a file name with a transaction code, so that all users of a particular transaction code will use the same VSAM file, while other users will access a different file. This statement is not necessary if 'WNDOFIL' is the only VSAM file in use, even though different transaction codes may be defined.

To display the File table, from the Auxiliary Functions menu key a "2" and press ENTER. This screen will appear as follows:



You can change any value where the cursor will stop when the TAB key is pressed (except the PF key line). The following discussion describes each field.

FIELDS OF THE FILE TABLE

FILE NAME — This field defines the file name to be associated with the corresponding transaction code specified in the one of the TRAN CODE fields. Values are:

symbol — Any valid VSAM file name. This should be the VSAM file that was created during the installation process.

[Note]: Multiple TRAN CODE fields on any one line of the table may be coded. For instance, if you wish to have both menu and message definitions in the same file, you may code the menu transaction ID, the message transaction ID and the file name on the same line.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

[Note]: As described in the *INSTALLATION* section, using alternate transaction codes for CICS-WINDOWS does not require the definition of those transactions anywhere other than in the CICS PCT. The File table is only required if you want to use alternate VSAM file name(s).

WINDOW TRAN CODE

This field is used to define the transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

symbol A valid transaction code that is used to invoke the sessions-management facilities of CICS-WINDOWS.

[Note]: The transaction code 'WNDO' does not need to be defined in a TRAN operand unless you intend to associate a file name other than WNDOFIL with it. In fact, by using alternate transaction codes, you do not need to define the transaction code 'WNDO' to CICS at all, if preferred.

MENU TRAN CODE

This field is used to define the Menu Definition transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

symbol A valid transaction code that is used to invoke the Menu Definition facility of CICS-WINDOWS.

MESSAGE TRAN CODE

This field is used to define the Message Broadcasting transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

symbol A valid transaction code that is used to invoke the Message Broadcast facility of CICS-WINDOWS.

HELP TRAN CODE

This field is used to define the transaction code that is to be associated with the file name specified in the corresponding FILE NAME field. Values are:

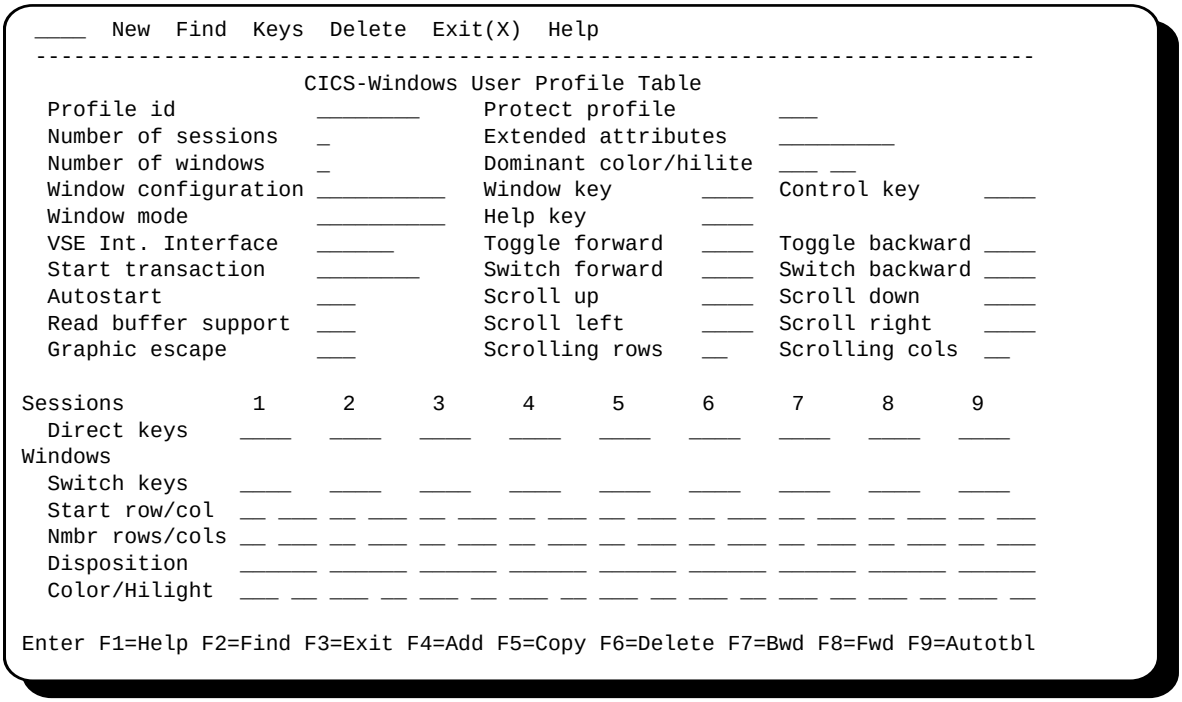
symbol A valid transaction code that is used to invoke the help maintenance facilities of the HELP-WINDOWS feature.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE USER PROFILE TABLE

The User Profile table is used to pre-define some or all of the start-up options for a given terminal, or for all terminals. This enables operators to have pre-defined CICS-WINDOWS options that differ from or augment the Global User Option settings.

To display the User Profile table, from the Auxiliary Functions menu key a "3" and press ENTER. This screen will appear generally as follows:



You can change any value where the cursor will stop when the TAB key is pressed (except the PF key line). The following discussion describes each field.

FIELDS OF THE USER PROFILE TABLE

AUTOSTART — This specifies whether this profile will automatically activate CICS-WINDOWS for an operator when they sign on to CICS. Valid entries are:

YES — Immediately after a CICS sign on is performed, CICS-WINDOWS will automatically perform a WNDO,ON command for operators with this profile.

NO — CICS-WINDOWS will not be automatically started. The operator must perform a WNDO,ON command.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CONTROL KEY

This is used to specify a PF or PA key to be pressed that will invoke the CICS-WINDOWS Control Window. For more information on the Control Window, see section 02 - OPERATION under THE CICS-WINDOWS CONTROL WINDOW.

DOMINANT COLOR/HILITE

If EXTENDED ATTRIBUTES is coded 'YES', 'COLOR' or 'HIGHLIGHT', this specifies the type of border that is to be used around the dominant, or active, window while in window mode.

Two fields are present. The first field is the color of the window border. Valid entries are:

(BLU)e, (GRE)en, (PIN)k, (RED), (TUR)quiose, (WHI)te, (YEL)low
The first three characters of the color that is desired.

NO The window borders of the dominant window are to display as the same color specified in the COLOR/HILITE field for that window.

The second field describes the type of highlighting to be used for the window. Valid entries are:

RB The window borders are to be displayed as reverse video.

IN The window borders are not to display (invisible).

NO The window borders are to remain normal.

EXTENDED ATTRIBUTES

If a terminal with this profile supports extended color and/or highlighting, this specifies the type of border that is to be used around each window when in window mode. Valid entries are:

NO The window borders are to display as vertical bars and underscores. Extended color and reverse video will **not** be used.

COLOR The window borders will assume the colors specified in the WINDOWS COLOR/HILIGHT fields. Reverse video borders will **not** be used.

HIGHLIGHT The window borders are to display in reverse video. Extended color will **not** be used.

YES The window borders are to display in reverse with the colors specified in the WINDOWS COLOR/HILIGHT fields.

[Note]: For extended attribute borders to be used, the CICS TCT definition must specify support for EXTENDEDDS, COLOR, and/or HILIGHT. If the TCT does not have these values coded, but you believe the terminal will support extended datastreams, you can set this same option on the User Configuration screen while in window mode to view the effect of each choice. This will send an extended datastream without regard to the TCT specification, which can result in a PROG471 error if the terminal will not support extended attributes. If the PROG471 error occurs, press the Window key to exit window mode and change the option back to NO.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

GRAPHICS ESCAPE

If this terminal supports graphics escape sequences, this field may be used to specify whether graphics escape characters are to be used to display the window border as a solid line around each window. Valid entries are:

YES Use graphics escape characters to display the window borders.

NO Do not use graphics escape characters.

[Note]: For graphics escape borders to be used, the CICS TCT definition should specify PS as a FEATURE. If the TCT does not have this value coded, but you believe the terminal will support graphics escape characters, you can set this same option on the WNIT screen while in window mode to view the effect. WNIT will send graphics escape characters without regard to the TCT specification, which can result in erroneously displayed window borders or even a PROG471 error.

HELP KEY This is used to specify a PF or PA key to be used as the CICS-WINDOWS Help Access key. For more information on the Help Access key, see section 07 - *THE HELP-WINDOWS FEATURE*.

Valid entries are any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

NUMBER OF SESSIONS

This is the number of sessions (virtual terminals) that an operator with this profile will have. Note that if this value exceeds the value coded for MAX LOGICAL TERMINALS on the User Options table, the maximum value will be used.

If this operand is omitted, the terminal operator will be prompted for the number of virtual terminals at initialization. Valid entries are:

number A number from 2 to 9, inclusive.

NUMBER OF WINDOWS

This is the number of windows an operator with this profile will have.

If this field is omitted, the operator will be prompted for the number and configuration of windows. Valid entries are:

number A number from 2 to 9, inclusive. Note that with STANDARD or VARIABLE window modes, 4 windows is the maximum value.

PROFILE ID This is the ID of which this profile will be referenced on the Auto-Init table. Note that the Profile ID can not be changed once the profile is created. To accomplish this you must first copy this profile to the desired Profile ID, then delete the old Profile. Valid entries are:

symbol Any 1 to 8 character name.

Default profiles may be created through the use of a symbolic Profile ID: &OP, &TRM or &USERID [see *USING THE AUTO-INIT TABLE TO GENERATE PROFILES* later in this section].

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PROTECT PROFILE

The User Configuration Display allows operators to change their configuration, and to save those changes in their profile. However if multiple operators are sharing the same profile, saving these changes would affect all operators associated with the profile. The Protect Profile option disables this feature so that the profile cannot be altered from the User Configuration Display. Values are:

YES Do not allow operators to alter their profile(s) from the User Configuration Display.

NO Allow operators to alter their profile(s) from the User Configuration Display.

READ BUFFER SUPPORT

This is used to specify whether CICS-WINDOWS issues a terminal Read Buffer command when the toggle key is pressed or the window key to enter window mode. This applies only to operators with this profile, and is only valid if READ BUFFER is coded "YES" in the User Option Table. Certain non-IBM terminals will sometimes not support the Read Buffer command, causing ATNI abends when the toggle key is pressed. This is the case if you are using VTAM PASSTHRU for your terminal. Valid entries are:

YES CICS-WINDOWS will issue a Read Buffer command for operators of this profile if READ BUFFER is coded "YES" in the User Option Table.

NO CICS-WINDOWS will not issue a Read Buffer command. You need to operate with READ BUFFER=NO if any of the following conditions is true:

1). You want to use the VIEW function of the SYS display and always see the latest output screen regardless of when the operator last toggled.

2). You wish to improve the response time for remote terminals when a toggle occurs.

SCROLL DOWNThis field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLL LEFT This field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLL RIGHTThis field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SCROLL UP This field is used to designate PF or PA keys to be used for a Window Scrolling key.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

See section 04 - *USING THE WINDOWING FEATURE* under *WINDOW SCROLLING KEY OPERATION* for an explanation of the use of Window Scrolling keys.

SCROLLING COLUMNS

This is the number of columns to scroll left or right when the Left or Right command is issued (or Scrolling Key is pressed).

SCROLLING ROWS

This is the number of rows to Scroll up or down when the Up or Down command is issued (or a Scrolling Key is pressed).

SESSIONS DIRECT KEYS

Direct session toggle keys. If this feature is desired, code a PF or PA key for each virtual terminal present on this physical terminal. The keys correspond one for one with each virtual terminal. That is, the first key coded will transfer control directly to virtual terminal 1, the second to virtual terminal 2, etc.

You may omit one or more keys in the list by skipping that position. If this is done, the virtual terminal corresponding to that key position will not have a direct key assigned to it.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*). The key selected must not be the same as any other designated key.

See the section entitled *DIRECT SESSION KEY OPERATION* for an explanation of the use of Direct Session keys.

START TRANSACTION

This provides a means of starting a user transaction automatically in each virtual terminal by connecting to an entry in the Application Startup table. You may define the same or different transactions in each session, and you may define data to be passed to the application program to vary the operation in each session.

When used, the transaction specified for virtual terminal number 1 will automatically start when the WNDO,ON command is completed (the User Configuration screen normally produced by WNDO,ON is suppressed). The transactions corresponding to the remaining sessions will be automatically started the first time the operator toggles into that session. For more information, see *THE APPLICATION STARTUP TABLE*, later in this section.

To use automatic transaction starting, code the one to eight character APPL ID (as coded on the appropriate table entry on the Application Startup table) in this field.

If automatic transaction starting is not desired, omit this keyword.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SWITCH BACKWARD KEY

This is the PF or PA key to be used in window mode to move "backward" from the active window to the next lower window number. Each time the Switch-Backward key is pressed, control moves from the current window to the previous one in sequence until window 1 is reached, at which time control moves to window number (NUMBER OF WINDOWS). Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

SWITCH FORWARD KEY

This is the PF or PA key to be used in window mode to move "forward" from the active window to the next higher window number. Each time the Switch-Forward key is pressed, control moves from the current window to the next one in sequence until window number (NUMBER OF WINDOWS+1) is reached, at which time control moves to window number one. Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

TOGGLE BACKWARD KEY

Code the PF or PA key to be used to move "backward" from one virtual terminal to the next lower terminal number. Each time the Toggle-Backward key is pressed, control moves from the current virtual terminal to the previous one in sequence until terminal number 1 is reached, at which time control moves to virtual terminal number (NUMBER OF SESSIONS). Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY*).

TOGGLE FORWARD KEY

This is the PF or PA key to be used to move "forward" from one virtual terminal to the next higher terminal number. Each time the Toggle-Forward key is pressed, control moves from the current virtual terminal to the next one in sequence until terminal number (NUMBER OF SESSIONS) is reached, at which time control moves to virtual terminal number one.

If the Toggle-Forward key is not desired, omit this field, in which case, the operator will be prompted for the Toggle-Forward key at WNDO,ON time. Valid entries are:

Any PF or PA key or CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*).

VSE INT. INTERFACE

This operand establishes the method of operation for using multiple Interactive Interface selection panels with CICS-WINDOWS for DOS VSE users only. Valid entries are:

YES CICS-WINDOWS will duplicate the selection panel from the first session into all sessions and allow operation of the selection panel in all sessions with only one operator sign-on.

NO CICS-WINDOWS will allow the selection panel to be active in session 1 only. Transactions must be started using transaction codes in the other sessions.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SIGNON CICS-WINDOWS will require a different operator sign-on in each session and will support different selection panels in each session.

[Note]: If "YES" or "SIGNON" is coded for any profile, the WNDOVSP program must be installed.

For more information, see *RUNNING CICS-WINDOWS WITH THE DOS VSE INTERACTIVE INTERFACE* in the section entitled *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS.*

WINDOW CONFIGURATION

This is the initial window configuration that an operator with this profile will have. Valid entries are:

HORIZONTAL

When the number of windows is "2", they will be horizontal windows.

For popup window mode this specifies that the first two windows will appear in size and shape as the standard two horizontal window configuration. Note that more than two windows may be available.

VERTICAL When the number of windows is "2", they will be vertical windows.

For popup window mode this specifies that the first two windows will appear in size and shape as the standard two vertical window configuration. Note that more than two windows may be available.

POPUP This option applies to popup window mode only, and simply specifies that no standard window configuration is desired.

3MAIN This option applies to popup window mode only, and specifies that the first three windows will appear in size and shape as the standard three window configuration. Note that more than three windows may be available.

4MAIN This option applies to popup window mode only, and specifies that the first four windows will appear in size and shape as the standard four window configuration. Note that more than four windows may be available.

WINDOW KEY This field is used to designate a PF or PA key to be used for entering and exiting window mode. If this field is omitted, the operator will be prompted with *WINDOWING DESIRED?* and, based on their response, prompted for the Window key. Valid entries are:

NO Windowing is not to be supported for this terminal, and the operator will not be prompted with *WINDOWING DESIRED?*.

YES The operator will be prompted for the window key and the prompt for *WINDOWING DESIRED?* will be skipped. Valid entries are:

PFxx Any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY*).

WINDOW MODE

This is the window mode to be used when initially entering window mode. Values for this field are:

STANDARD This will set the terminal to use Standard window mode. In this mode, the first time the window key is pressed to enter window mode, no window is designated active and all windows are unprotected.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

VARIABLE

This will set the terminal to use Variable or Dominant window mode (see *VARIABLE WINDOW CONFIGURATIONS* in the section entitled *USING THE WINDOWING FEATURE*). In this mode, the first time the window key is pressed to enter window mode, window number 1 will be the dominant active window and all other windows will be protected.

POPUP This will set the terminal to use Popup window mode. This mode is much like VARIABLE window mode except that all windows are free floating. That is, any window can appear anywhere on the screen.

WINDOWS COLOR/HILITE

If EXTENDED ATTRIBUTES is coded 'YES', 'COLOR' or 'HIGHLIGHT', this specifies the type of border that is to be used around the window when in window mode. Note that this is only valid for POPUP window mode.

Two fields are present. The first field is the color of the window border. Valid entries are:

(BLU)e, (GRE)en, (PIN)k, (RED), (TUR)quiose, (WHI)te, (YEL)low

The first three characters of the color that is desired.

The second field describes the type of highlighting to be used for the window. Valid entries are:

RB The window borders are to be displayed as reverse video.

RW While the window is recessive, the entire window will be displayed as reverse video. If extended highlighting is not in effect, the text in the window will display as the color of the window borders.

IN The window borders are not to display (invisible).

NO The window borders are to remain normal.

WINDOWS DISPOSITION

These fields determine how a window is displayed while it is not the dominant window. Note that DISPOSITION is only valid for POPUP window mode.

FULL The window will display as a full screen in the background.

HIDE The window will not appear on the screen.

ICON The window will appear at the bottom of the screen as an icon, showing the current ID of the transaction that is operating in that window.

NORMAL The window will remain in place and may be overlapped by the dominant window.

WINDOWS NUMBER ROWS/COLS

These fields are used to specify the size of each window.

To use this feature, code the number of rows and columns.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WINDOWS START ROW/COL

These fields are used to specify the start position of the upper left corner of each window.

To use this feature, code the row and column position relative to "1". Coding zeros for any window will cause CICS-WINDOWS to choose the location of the window. If a Full Screen window is desired, enter an "F" in this field.

WINDOWS SWITCH KEYS

If this feature is desired, code a PF or PA key for each window, as desired. The keys correspond one for one with each window. That is, the first key coded will do a Window Switch command to window 1, making it the Dominant window, the second to window 2, etc. You may code as many keys as the available number of windows.

You may omit one or more keys in the list by skipping that field. If this is done, the window corresponding to that key position will not have a Window Switch key assigned to it.

Valid keys are any PF or PA key and CSEL (see *USING THE CURSOR-SELECT KEY AS A HOT KEY* in section 12 - *SPECIAL CONSIDERATIONS*). The key selected must not be the same as any other designated key.

See the section entitled *WINDOW SWITCH KEY OPERATION* for an explanation of the use of Window Switch keys.

ENTERING COMMANDS AT THE USER PROFILE TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help window. If the cursor is positioned to an unprotected field, the help will pertain to that field, otherwise the help will pertain to the User Profile display.

PF2 FIND Initiate the FIND function, to display a different profile.

PF3 EXIT Return to the Auxiliary Functions menu.

PF4 ADD Add a new profile entry to the table.

PF5 COPY Copy this profile. This may be used when creating a new profile if it is advantageous to copy an existing profile, then later modify any differing fields as desired.

PF6 DELETE Delete this profile entry from the table.

PF7 BWD Browse backward to the previous table entry.

PF8 FWD Browse forward to the next table entry.

PF9 AUTOTBL Display the Auto-Init table. (Same as selection 4 from the Auxiliary Functions menu.)

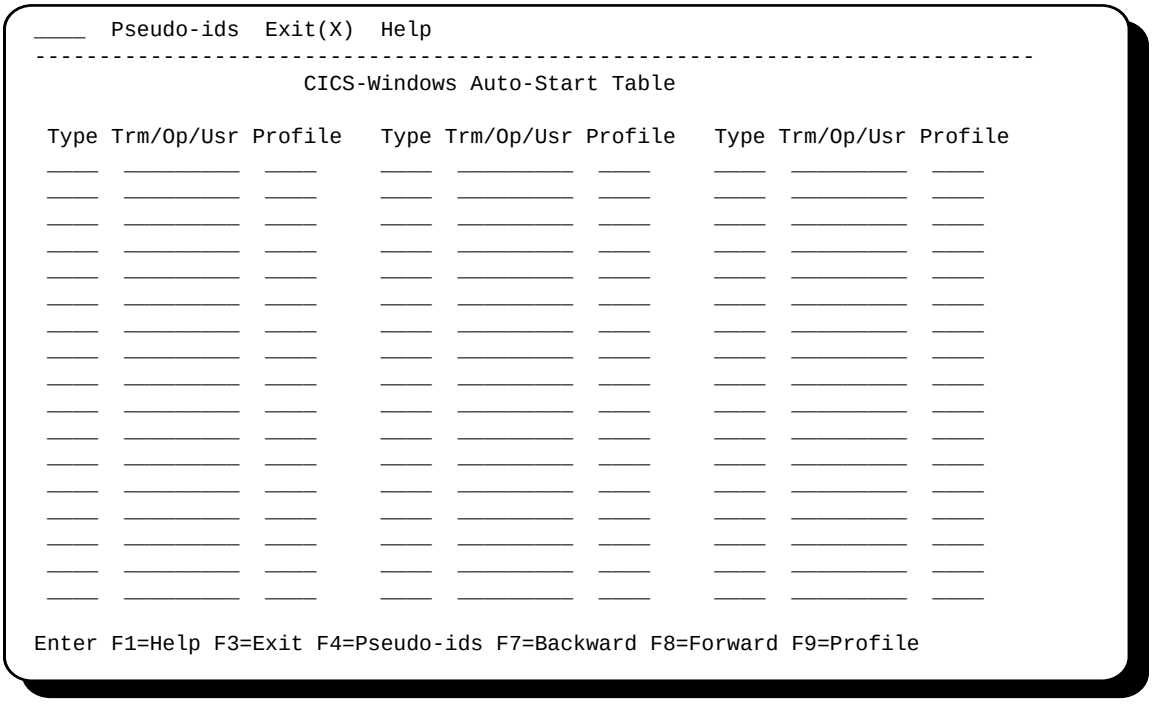
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE AUTO-INIT TABLE

The Auto-Init table is used by CICS-WINDOWS to associate an operator with a User Profile when a WNDO,ON command has been issued by that operator. In addition, this table can also be used to assign preselected pseudo terminal IDs to that operator.

When performing a WNDO,ON, CICS-WINDOWS will use the profile of the first matching auto-init table entry; thus the relative position of auto-init table entries is crucial in determining the profile to be used. For instance, if a completely generic auto-init entry (solo asterisk) is the first entry in the table, that profile will be used for **all** operators that issue a WNDO,ON command. Usually, any generic or wildcard entries should appear at the bottom of the list.

To display the Auto-Init table, from the Auxiliary Functions menu key a "4" and press ENTER. This screen will appear generally as follows:


____ Pseudo-ids Exit(X) Help
CICS-Windows Auto-Start Table
Type Trm/Op/Usr Profile Type Trm/Op/Usr Profile Type Trm/Op/Usr Profile
Enter F1=Help F3=Exit F4=Pseudo-ids F7=Backward F8=Forward F9=Profile

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE AUTO-INIT TABLE

PROFILE — This is the ID of the User Profile that is to be used for this table entry. Valid entries are any previously defined User Profile.

PSEUDO IDS — These are the pseudo terminal Ids that are connected to each table entry. You may add new Ids or modify existing Ids in these fields. These fields may be coded in the following ways:

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

1). Code the explicit unique terminal IDs to be assigned to each of the virtual terminals. Code one 4-character ID per virtual terminal. Each ID must be unique, that is, it must not be present either in the Terminal Control Table (TCT) or specified as a Pseudo ID in any other entry.

The first terminal uses the real terminal ID and is not present in the list. Thus, if NUMBER OF SESSIONS is specified as '4', the number of IDs to be coded is three. If NUMBER OF SESSIONS is omitted, the number of Pseudo IDs to be coded must equal the value coded for MAXIMUM VIRTUAL TERMINALS minus 1.

[Note]: You should not code specific terminal IDs unless the TYPE field specifies TERM and the Terminal ID is fully defined (non-generic).

2). If Pseudo IDs are to be automatically generated by CICS-WINDOWS, code '*GEN' in the first pseudo ID field. This will cause WINDOWS to generate unique pseudo IDs and bypass the pseudo ID prompt at initialization for the terminal.

When Pseudo IDs are generated by CICS-WINDOWS, the real terminal ID is used as the basis for constructing a unique ID for each virtual terminal. Each position of the real ID is copied to the corresponding position in the pseudo ID, then one character of the ID is set to uppercase or lowercase, or to a special character (see following illustration). The result is a unique ID for each virtual terminal that still resembles the real terminal ID.

The following illustration shows the position of the lowercase characters for each of the eight possible virtual terminals. (U=uppercase, L=lowercase).

Real terminal ID	UUUU	Example:	TRMX
Virtual terminal 1	UUUL	TRMx	
Virtual terminal 2	UULU	TRmX	
Virtual terminal 3	ULUU	TrMX	
Virtual terminal 4	LUUU	tRMX	
Virtual terminal 5	LUUL	tRMx	
Virtual terminal 6	LULL	tRmx	
Virtual terminal 7	LLLL	trmx	
Virtual terminal 8	ULLL	Trmx	

When numeric characters are present in the real ID, some models of 3270 terminals will not properly display the lowercase numeric characters (3179, 3180 terminals). This does not cause any significant problems, but if you prefer not to use lowercase numeric, select SPECIAL CHARACTER REQUEST in the User Options table. This specifies the use of special characters in place of lowercase numeric in the terminal ID.

The special characters used are the characters corresponding to the shift position of most 3270 keyboards. They are:

'1'	=	'|'	(vertical bar)
'2'	=	'@'	(at sign)
'3'	=	'#'	(pound sign)
'4'	=	'$'	(dollar sign)
'5'	=	'%'	(percent sign)
'6'	=	'¢'	(cent sign)
'7'	=	'&'	(ampersand)
'8'	=	'*'	(asterisk)
'9'	=	'('	(left parenthesis)
'0'	=	')'	(right parenthesis)

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

You can also write your own user exit program to generate the pseudo IDs using any method that you prefer. See the discussion of *PSEUDO TERMINAL ID GENERATION EXIT* in section 15 - *USER EXITS AND PROGRAM INTERFACES.*

3). If Pseudo Terminal IDs are not to be assigned for this terminal code "NO" in the first field. This will cause each virtual terminal to use the real terminal ID.

4). If these fields are left blank, the operator will be prompted for the Pseudo IDs unless BYPASS PSEUDO IDS is selected in the User Option Table.

[Note]: If a pseudo ID generation exit is to be used, you must code “*GEN” in the first field.

START Code YES in this field to activate AUTO-START for this entry. With Auto-Start, CICS-WINDOWS will be automatically activated at the terminal when the operator signs on to CICS. Note that if your sign-on transaction is other than CESN or CSSN, you may need to define additional sign-on Transactions to CICS-WINDOWS (see *SIGNON TRANSACTION TABLE*).

Code No in this field (or omit) if you do not wish to use Auto-Start. In this case, CICS-WINDOWS must be activated by a WNDO,ON command, either explicitly or under program control.

TRM/OP/USR

This field is used to supply the actual data that CICS-Windows will use to identify an operator. Its meaning depends on the value specified for the TYPE field, described following.

1) Code an individual statement for each terminal to be defined, coding the TERM keyword as the four-character terminal ID of the terminal to be configured.

2) Define a range of terminals to be configured using the parameters specified on this statement. Code the TERM operand as two terminal IDs separated by a hyphen. With this method, when a WNDO,ON command is performed at any terminal which has a terminal ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will use the profile of this statement.

3) Define a generic definition using "wild-card" characters. Code the TERM operand with question marks (?) in one or more positions of the terminal ID. With this method, when a WNDO,ON command is performed at any terminal, if all positions in the terminal ID match the corresponding positions in the TERM operand where question marks are not coded, that terminal will use the profile of this statement.

4) Define a single statement that applies to all terminals. Code the TERM operand as "*". With this method, this statement applies to any terminal using CICS-WINDOWS which is not explicitly identified by another statement.

If a solo asterisk is coded and other statements are present, this entry should be the last statement. This allows some terminals to be defined explicitly, using different start-up options than the generic entry.

For TYPE=OP

Code the operator name, as it appears in the TCT. This is the three-character CICS operator ID of the operator for which the configuration described by the corresponding profile applies. The operator ID keyword may be coded in any of the following four ways:

1) Code an individual statement for each operator to be defined, coding the three-character operator ID of the user to be configured.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

2) Define a range of operators to be configured using the parameters specified on this statement. Code the two operator IDs separated by a hyphen. With this method, when a WNDO,ON command is performed at any terminal which has an operator ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will use the profile on this statement.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

3) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the operator ID. With this method, when a WNDO,ON command is done at any terminal, if all positions in the operator ID match the corresponding positions where question marks are not coded, that terminal will use the profile on this statement.

4) Define a single statement which applies to all users. Code the field as "*". With this method, this statement applies to any terminal using CICS-WINDOWS which is not explicitly identified by another statement.

If a solo asterisk is coded and other statements are present, this entry should be the last statement. This allows some users to be defined explicitly, using different start-up options than the generic entry.

For TYPE=USER

Code the user ID, as it appears in the SNT. This is the eight-character CICS user ID of the operator for which the configuration described by the corresponding profile applies. The user ID Keyword may be coded in any of the following ways:

1) Code an individual statement for each user to be defined, coding the eight-character user ID of the user to be configured.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the user ID. With this method, when a WNDO,ON command is done at any terminal, if all positions in the user ID match the corresponding positions where question marks are not coded, that terminal will use the profile on this statement.

3) Define a single statement which applies to all users. Code the field as "*". With this method, this statement applies to any user using CICS-WINDOWS which is not explicitly identified by another statement.

If a solo asterisk is coded and other statements are present, this entry should be the last statement. This allows some users to be defined explicitly, using different start-up options than the generic entry.

TYPE This field is used to specify the type of data CICS-WINDOWS is to use to determine what profile to assign to an operator when a WNDO,ON command is issued. Valid entries are:

TERM This table entry is to associate the 4-character terminal ID with a User Profile

OP This table entry is to associate the 3-character operator identifier with a User Profile.

USER This table entry is to associate the 8-character user identifier with a User Profile. This option is only valid for CICS releases 1.7 and later.

In addition to the entries that may be coded in this field, there are two commands that may be keyed in this field:

- The (D)elete command will delete an entry and shift all lower entries up into the vacated slot.
- The (I)nsert command will shift all lower entries down (including the entry in which the command was issued), which will create a vacant slot for inserting a new auto-init entry.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER — Apply any changes that you have made to the table.

PF1 HELP — Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT — Return to the Auxiliary Functions menu.

PF4 AUTOINIT
Remove the Pseudo IDs data and view all Auto-Init entries in a 2-up display. Pressing PF4 again will return to the 1-up display showing Pseudo Ids

PF7 BACKWARD
Browse backward to the previous page of table entries.

PF8 FORWARD — Browse forward to the next page of table entries.

PF9 PROFILE — Display the Profile for the entry that the cursor is pointing when PF9 is pressed. (Same as selection 2 from the Auxiliary Functions menu.)

In addition to the PF key functions, there are two commands that may be keyed in the TYPE field:

- The (**D**)elete command will delete an entry and shift all lower entries up into the vacated slot.
- The (**I**)nsert command will shift all lower entries down (including the entry in which the command was issued), which will create a vacant slot for inserting a new auto-init entry.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Customization may be performed dynamically, bypassing the Auxiliary Functions transaction WAUX. The Auto-Init table may be used to associate operators or terminals with symbolic User Profile tables, which may then be customized by the user and saved, creating a new User Profile table with a Profile ID generated from the symbolic Profile ID.

If no specific profile is found for an operator (designated by the Operator or User ID) or terminal, CICS-WINDOWS selects a default profile specified in the Auto-init table. If the default profile specified is symbolic, the operator may then alter any of the customization options (except the Profile ID) and save the profile by pressing PF4. A new User Profile table is dynamically created, and subsequent WNDO,ON operations will associate the profile with the operator or terminal, just as if a profile had been created through the WAUX transaction as described in the topic entitled The *User Profile Table* earlier in this section. CICS-WINDOWS will use information supplied by CICS to construct a profile ID based on the terminal ID, Operator ID, or User ID as specified by the Profile ID in the symbolic User Profile which serves to associate the operator or terminal with the profile on subsequent invocations of CICS-WINDOWS.

A symbolic profile is created by specifying one of the following three types in the Profile ID field of the User Profile Table, depending on the desired association of the saved profile:

&TRM The saved profile will be associated with a terminal ID.
&OP The saved profile will be associated with the Operator ID.
&USERID The saved profile will be associated with the User ID.

For example, a profile using the symbolic profile ID '&USERID' might appear as follows:

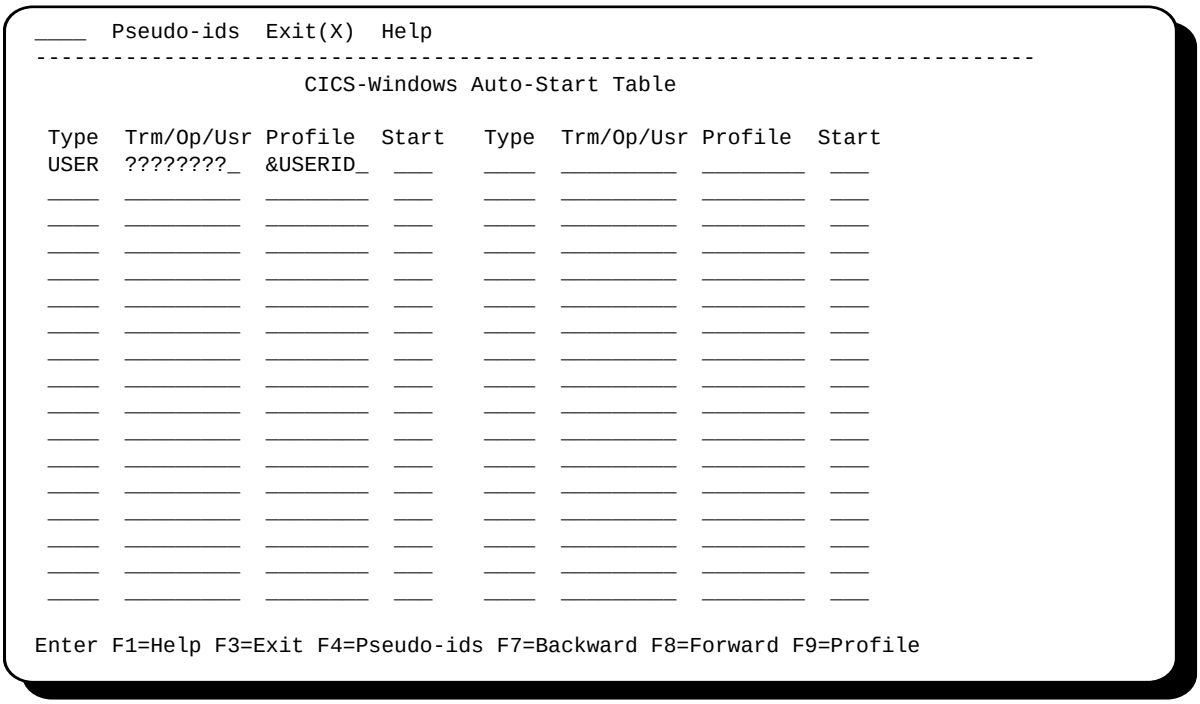
```
____  New  Find  Keys  Delete  Exit(X)  Help
--------------------------------------------------------------------------------
                     CICS-Windows User Profile Table
  Profile id          &USERID_     Protect profile        NO_

  Number of windows   4            Dominant color/hilite  NO_ RB
  Window configuration POP-UP____  Window key       PF22  Control key     ____
  Window mode         POP-UP____   Help key         ____
  VSE Int. Interface  NO____       Toggle forward   PF23  Toggle backward PF24
  Start transaction   ________     Switch forward   PF22  Switch backward PF21
  Autostart           NO_          Scroll up        ____  Scroll down     ____
  Read buffer support YES          Scroll left      ____  Scroll right    ____
  Graphic escape      NO_          Scrolling rows   __    Scrolling cols  __

Sessions          1      2      3      4      5      6      7      8      9
  Direct keys    ____   ____   ____   ____   ____   ____   ____   ____   ____
Windows
  Switch keys    ____   ____   ____   ____   ____   ____   ____   ____   ____
  Start row/col  01 001 05 003 10 006 15 009 __ ___ __ ___ __ ___ __ ___ __ ___
  Nmbr rows/cols 14 065 14 065 14 065 14 065 __ ___ __ ___ __ ___ __ ___ __ ___
  Disposition    NORMAL NORMAL NORMAL NORMAL  ______ ______ ______ ______ ______
  Color/Hilight  YEL RW TUR RW PIN RW RED RW ___ __ ___ __ ___ __ ___ __ ___ __

Enter F1=Help F2=Find F3=Exit F4=Add F5=Copy F6=Delete F7=Bwd F8=Fwd F9=Autotbl
```

This symbolic profile may be associated with specific or generic terminals or operators in the Auto-init table, as described previously in this section under the topic *THE AUTO-INIT TABLE*. The use of

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

wild-card characters allows symbolic profiles to be defined as default profiles which will be retrieved whenever no specific profile is found for a terminal or operator.

The following Auto-init table is an example of designating a symbolic profile as the default profile in all circumstances where a specific profile is not associated with a User ID:


____ Pseudo-ids Exit(X) Help
CICS-Windows Auto-Start Table
Type Trm/Op/Usr Profile Start Type Trm/Op/Usr Profile Start
USER ????????_ &USERID_ ___ ____ _____ _____ ___
Enter F1=Help F3=Exit F4=Pseudo-ids F7=Backward F8=Forward F9=Profile

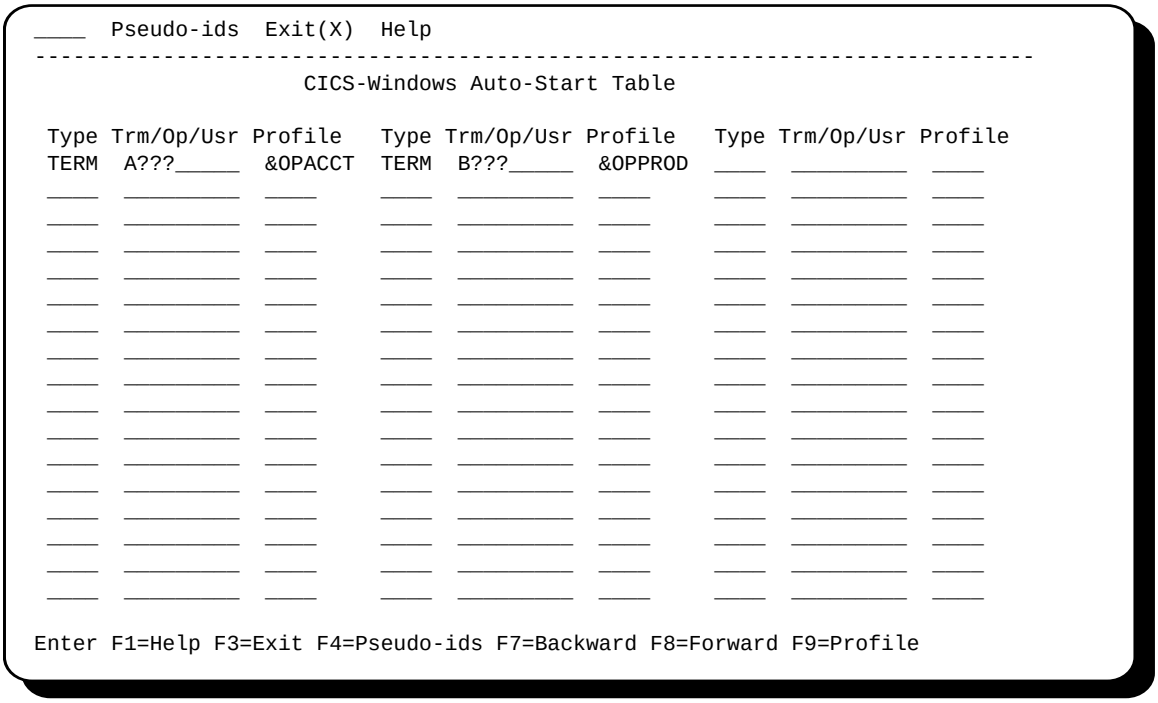
By creating a default profile with the Profile ID '&USERID', an operator with the User ID 'JOHN' may perform a WNDO,ON and, if no previously defined profile is found in the Auto-init table for the operator or terminal, be assigned the default User Profile with the Profile ID '&USERID'. The operator may then customize the User Profile table when it appears and save it by pressing PF4. The next time the operator activates CICS-WINDOWS, the Auto-init table is scanned for 'JOHN'. No explicit entry has been made in the Auto-init table, but the Auto-init entry specifying the symbolic profile ID '&USERID' causes CICS-WINDOWS to search for a User Profile table associated with the User ID. Since the operator saved the profile, CICS-WINDOWS retrieves the User Profile table which has dynamically been created with the Profile ID 'JOHN'.

The dynamically generated profile may be viewed and modified, if desired, by selecting the 'Profile table' option from the Auxiliary Functions menu. The symbolic profile ID is replaced with the operator ID, user ID, or terminal id supplied by CICS, depending on which type is specified. In the above instance, 'JOHN', the user ID became the Profile ID for the dynamically created User Profile table since the symbolic profile specified &USERID.

Multiple versions of symbolic profiles may created by appending characters to the end of the symbolic Profile ID. Since Profile IDs may contain a maximum of ten characters, 7 characters may be appended to the end of &OP, 6 to the end of &TRM and 2 to the end of &USERID. The appended characters are added to the operator, user, or terminal ID whenever a User Profile table is dynamically created. In the above example, replacing the symbolic profile &USERID with &USERIDA1 would have generated the dynamically created the User Profile table JOHNA1.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Using multiple versions of symbolic profiles and generic associations in the Auto-init table allow a high degree of dynamic customization. For instance, &OPACCT may be used to associate a particular profile with terminals in the accounting department while &OPPROD is used to associate another profile with terminals in the production department. These specific associations are possible through the use of generic characters in the 'Trm/Op/Usr' field of the Auto-init Table. For example, assume all terminals in the accounting department use 'A' as the first character of the terminal id, and all terminals in the production department use 'P' as the first character. The following entries in the Auto-init Table would cause CICS-WINDOWS to select the &OPACCT profile if an operator or terminal with no previous Auto-init specification activated CICS-WINDOWS from a terminal in the accounting department, and the &OPPROD profile from a terminal in the production department:


____ Pseudo-ids Exit(X) Help
CICS-Windows Auto-Start Table
Type Trm/Op/Usr Profile Type Trm/Op/Usr Profile Type Trm/Op/Usr Profile
TERM A???_____ &OPACCT TERM B???_____ &OPPROD
Enter F1=Help F3=Exit F4=Pseudo-ids F7=Backward F8=Forward F9=Profile

It is important to reiterate that the type of default (&OP, &USERID, &TRM) is associated with the type of profile CICS-WINDOWS saves when an operator chooses to save the default profile and searches for when subsequent activations occur. The above example illustrates that the type of default profile retrieved initially and the type saved from the default are not necessarily identical. If an operator in the above example chooses to save a profile, that profile will be associated with the operator ID supplied by CICS; if &TRM had been specified, the saved profile would have been invoked whenever CICS-WINDOWS was invoked from the terminal, which would defeat the purpose of specifying separate default profiles for the accounting and production departments.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Assuming an operator with the operator ID 'ABC' invokes CICS-WINDOWS from a terminal in the accounting department, modifies the default profile and saves it, the following profile would be retrieved using the WAUX transaction:

```
 ____  New  Find  Keys  Delete  Exit(X)  Help
 ------------------------------------------------------------------------------
                       CICS-Windows User Profile Table
  Profile id           ABCACCT_     Protect profile        NO_
  Number of sessions   9            Extended attributes    YES_____
  Number of windows    4            Dominant color/hilite  NO_ RB
  Window configuration POP-UP____   Window key       PF22  Control key     ____
  Window mode          POP-UP____   Help key         ____
  VSE Int. Interface   NO____       Toggle forward   PF23  Toggle backward PF24
  Start transaction    ________     Switch forward   PF22  Switch backward PF21
  Autostart            NO_          Scroll up        ____  Scroll down     ____
  Read buffer support  YES          Scroll left      ____  Scroll right    ____
  Graphic escape       NO_          Scrolling rows   __    Scrolling cols  __

Sessions          1      2      3      4      5      6      7      8      9
  Direct keys    ____   ____   ____   ____   ____   ____   ____   ____   ____
Windows
  Switch keys    ____   ____   ____   ____   ____   ____   ____   ____   ____
  Start row/col  01 001 05 003 10 006 15 009 __ ___ __ ___ __ ___ __ ___ __ ___
  Nmbr rows/cols 14 065 14 065 14 065 14 065 __ ___ __ ___ __ ___ __ ___ __ ___
  Disposition    NORMAL NORMAL NORMAL NORMAL  _______ _______ _______ _______ _______
  Color/Hilight  YEL RW TUR RW PIN RW RED RW ___ __ ___ __ ___ __ ___ __ ___ __

Enter F1=Help F2=Find F3=Exit F4=Add F5=Copy F6=Delete F7=Bwd F8=Fwd F9=Autotbl
```

Note that the ampersand and 'OP' characters of the symbolic profile ID have been replaced by the operator ID and the identifier appended. This profile may now be modified, copied or otherwise manipulated as any profile created through the WAUX transaction.

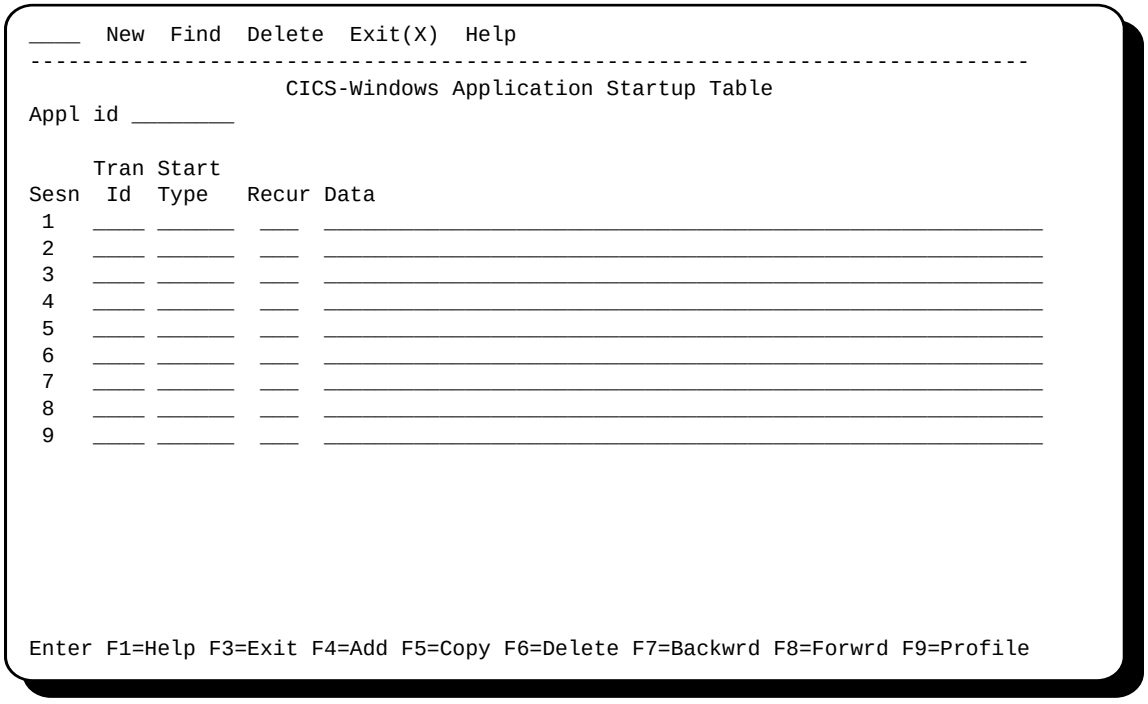
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE APPLICATION STARTUP TABLE

This table provides a means of starting a user transaction automatically in each virtual terminal. You may define the same or different transactions in each session, and you may define data to be passed to the application program to vary the operation in each session.

When used, the transaction specified for virtual terminal number 1 will automatically start when the WNDO,ON command is completed. The transactions corresponding to the remaining sessions will be automatically started the first time the operator toggles into that session.

To display the Application Startup table, from the Auxiliary Functions menu key a "5" and press ENTER. This screen will appear generally as follows:


____ New Find Delete Exit(X) Help
CICS-Windows Application Startup Table
Appl id ________
Tran Start
Sesn Id Type Recur Data
1 ____ _____ ___ _____
2 ____ _____ ___ _____
3 ____ _____ ___ _____
4 ____ _____ ___ _____
5 ____ _____ ___ _____
6 ____ _____ ___ _____
7 ____ _____ ___ _____
8 ____ _____ ___ _____
9 ____ _____ ___ _____
Enter F1=Help F3=Exit F4=Add F5=Copy F6=Delete F7=Backwrd F8=Forwrd F9=Profile

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE APPLICATION STARTUP TABLE

APPL ID — This is the name of this application startup table. This is the name that is coded in the START TRANSACTION field on a User Profile. In other words, this is the link between a profile and this Start Transaction definition. Values are:

Any valid 1 to 8 character ID.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DATA If START TYPE=ATTACH, the DATA keyword defines the data to be passed to the application program in the terminal I/O area. The data defined is the data which would be entered immediately following the transaction code if this transaction were entered on a terminal screen. Thus, if only the trancode is required as input, the DATA keyword should be omitted.

Example: To automatically start a transaction which would normally be entered on the screen in the form:

FSRV I ACCTFIL A*

the following coding would be required in the statement:

TRANSACTION ID	FSRV
START TYPE	ATTACH
RECUR	YES or NO
DATA	_I_ACCTFIL_A*

[Note]: Automatic initiation of transactions occurs either in full-screen mode when a toggle is performed, or in window mode when a toggle or window switch is performed.

RECUR This specifies whether the transaction to be started in this session will be initiated every time the session is toggled into or only the first time.

This feature is designed as an aid in integrating session applications. By initiating a transaction every time a session is entered, you can cause things to happen automatically when the operator presses the toggle key.

For instance, the transaction in session number 1 could write a record to temporary storage saving the control information for a given file. A recurring transaction could be specified in session number 2 which, upon invocation, would retrieve the saved control information, fetch the data record and display some related application information.

A special interface to CICS-WINDOWS is provided which allows you to change the transaction to be initiated in the recurring session under program control. This interface is described in the section entitled *USER EXITS AND PROGRAM INTERFACES*.

YES The transaction will be initiated every time the user toggles into this session.

NO The transaction will be initiated only upon the first toggle into this session.

The default is “NO.”

SESN This is the session, or virtual terminal number where this transaction is to be initiated.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

START TYPE This is the technique for CICS-WINDOWS to use when starting this transaction. Valid entries are:

ATTACH This means that the transaction will be started in the same way that CICS starts a transaction when the transaction code is entered from the terminal. That is, an I/O area is passed to the application containing the transaction code in the first position and any associated data following. ATTACH must be used if data is to be passed to the application, or if the transaction being started runs in a remote region through MRO or ISC.

START START means that the transaction will be initiated using an EXEC CICS START command. When using START, no terminal I/O area is passed to the application program. You can not pass any data to the application program when using the START operand.

[Note]: There is no "preferred" method here. If a transaction does not need incoming data, either method will work. ATTACH will be slightly quicker since it avoids the CICS Automatic Transaction Initiation mechanism.

TRANSATION ID

This is the transaction code to be automatically started in the corresponding session number.

Any valid 1 to 4 character transaction code that is defined to CICS in the PCT.

ENTERING COMMANDS AT THE APPLICATION STARTUP TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information pertaining to the field to which the cursor pointed when the PF key was pressed.

PF2 FIND Invoke the FIND function to display another Application Startup entry.

PF3 EXIT Return to the Auxiliary Functions menu.

PF4 ADD Add a new entry to the table.

PF5 COPY Copy the displayed entry. This may be used when creating a new table entry if it is advantageous to copy an existing entry, then later modify any differing fields as desired.

PF6 DELETE Delete the displayed entry from the table.

PF7 BACKWARD Browse backward to the previous Application Startup table

PF8 FORWARD Browse forward to the next Application Startup table.

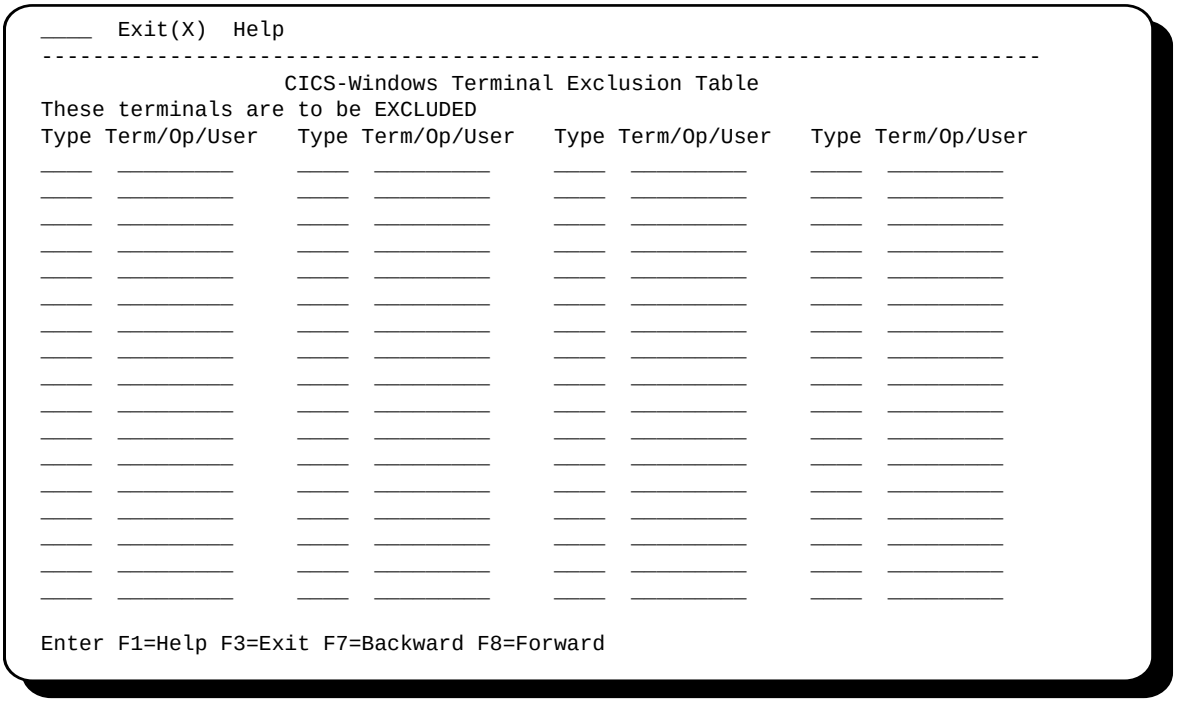
PF9 PROFILE Display the Profile table. (Same as selection 3 from the Auxiliary Functions menu.)

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE TERMINAL EXCLUSION TABLE

The Terminal Exclusion table is used to define one or more terminal IDs or operators which are to be excluded from using CICS-WINDOWS or included as the only terminals which can use CICS-WINDOWS, according to the specification at the top of the screen.

To display the Terminal Exclusion table, from the Auxiliary Functions menu key a "6" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Terminal Exclusion Table
These terminals are to be EXCLUDED
Type Term/Op/User Type Term/Op/User Type Term/Op/User Type Term/Op/User
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE TERMINAL EXCLUSION TABLE

THESE TERMINALS ARE TO BE
This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED
All entries on this table will not be allowed to use CICS-WINDOWS.

INCLUDED
Only the entries appearing on this table will be allowed to use CICS-WINDOWS.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TERM/OP/USER

This field is used to supply the actual data that CICS-WINDOWS will use to match with the operator that is attempting to perform a WNDO,ON command. Its meaning depends on the value specified for the TYPE field, described following.

For TYPE=TERM

Code the terminal ID, as it is known to CICS. This keyword may be coded in any of the following ways:

1) Code an individual statement for each terminal to be defined, coding the TERM keyword as the four-character terminal ID of the terminal to be excluded or included.

2) Define a range of terminals to be configured using the parameters specified on this statement. Code the TERM operand as two terminal IDs separated by a hyphen. With this method, when a WNDO,ON command is performed at any terminal that has a terminal ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will be excluded or included.

3) Define a generic definition using "wild-card" characters. Code the TERM operand with question marks (?) in one or more positions of the terminal ID. With this method, when a WNDO,ON command is issued at any terminal, if all positions in the terminal ID match the corresponding positions in the TERM operand where question marks are not coded, that terminal will be excluded or included.

For TYPE=OP

Code the operator name, as it appears in the TCT. This is the three-character CICS operator ID. The operator ID keyword may be coded in any of the following ways:

1) Code an individual statement for each operator to be defined, coding the three-character operator ID of the user to be excluded or included.

2) Define a range of operators to be configured using the parameters specified on this statement. Code the two operator IDs separated by a hyphen. With this method, when a WNDO,ON command is performed at any terminal that has an operator ID equal-to or greater than the first ID and equal-to or less than the second ID, that terminal will be excluded or included.

3) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the operator ID. With this method, when a WNDO,ON command is performed at any terminal, if all positions in the operator ID match the corresponding positions where question marks are not coded, that terminal will be excluded or included.

For TYPE=USER

Code the user ID, as it appears in the SNT or RACF profile. This is the eight-character CICS user ID. The user ID keyword may be coded in any of the following ways:

1) Code an individual statement for each user to be defined, coding the eight-character user ID of the user to be excluded or included.

2) Define a generic definition using "wild-card" characters. Code question marks (?) in one or more positions of the user ID. With this method, when a WNDO,ON command is performed at any terminal, if all positions in the user ID match the corresponding positions where question marks are not coded, that terminal will be excluded or included.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TYPE This field is used to specify the type of data CICS-WINDOWS is to use to determine if it should allow an operator to perform a WNDO,ON command. Valid entries are:

TERM This is a terminal table entry.

OP This is an operator table entry.

USER This is a User ID table entry. This option is only valid for CICS releases 1.7 and later.

ENTERING COMMANDS AT THE TERMINAL EXCLUSION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER Apply any changes that you have made to the table.

PF1 HELP Display a Help screen for information on the field that the cursor was in when the PF key was pressed.

PF3 EXIT Return to the Auxiliary Functions menu.

PF7 BACKWARD Browse backward to the previous page of table entries.

PF8 FORWARD Browse forward to the next page of table entries.

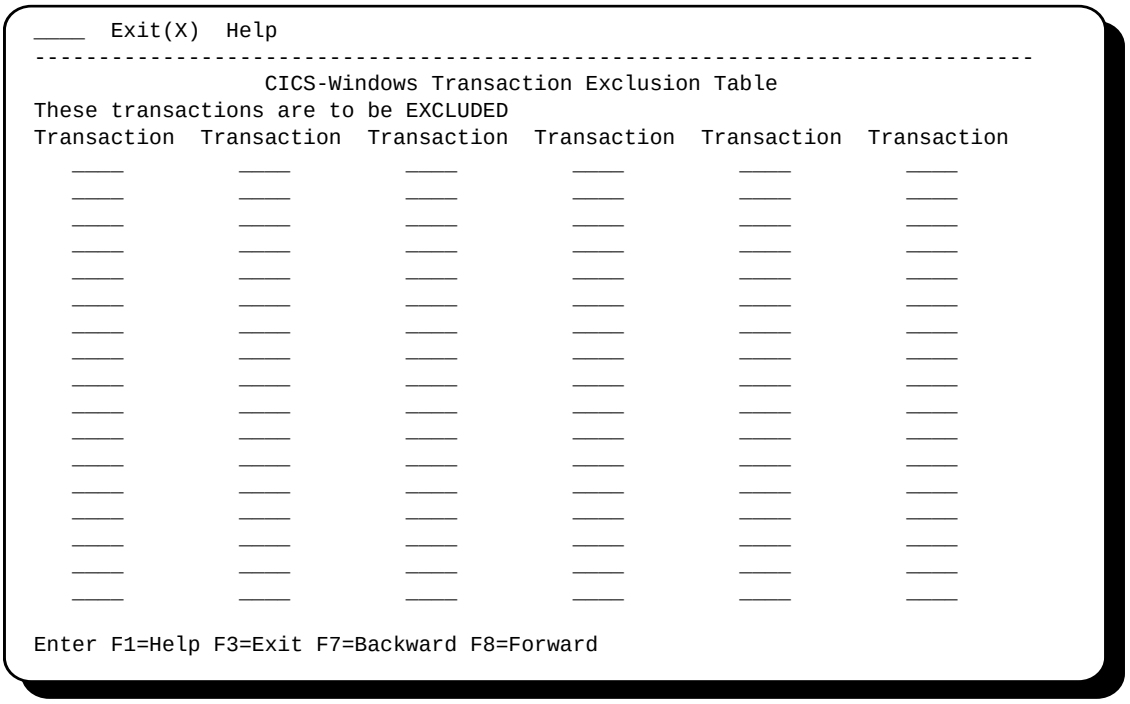
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE TRANSACTION EXCLUSION TABLE

An excluded transaction is one which overrides the “hot” PF keys when active. That is, if a toggle key, window key, switch key, control key or any of the PF key functions that can be assigned is pressed while the transaction is active, that PF key is passed directly to the transaction without performing the assigned CICS-WINDOWS function. Note that transaction exclusion does not preclude the use of toggling of windowing through the use of “transparent” commands with the control character. This is intended as a method of disabling hot keys for transactions that use all available keys.

This table is used to define one or more transaction codes which are to be excluded from recognizing the toggle keys or included as the only transactions that will recognize the toggle keys, according to the specification in the table.

To display the Transaction Exclusion table, from the Auxiliary Functions menu key a "7" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Transaction Exclusion Table
These transactions are to be EXCLUDED
Transaction Transaction Transaction Transaction Transaction Transaction
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE TRANSACTION EXCLUSION TABLE

THESE TRANSACTIONS ARE TO BE

This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED
All transactions on this table will not be allowed to toggle while using CICS-WINDOWS.

INCLUDED
Only the transactions appearing on this table will be allowed to toggle while using CICS-WINDOWS.

TRANSACTION
This is the transaction code as it is known to CICS. Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1) Code the full 4-character transaction code.

2) Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3) Define a generic definition using "wild-card" characters. Code the TRAN operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRAN operand where question marks are not coded, that transaction will be considered a match.

Do not code TRAN=* or TRAN=???? (all wild-cards). This will specify all transactions.

ENTERING COMMANDS AT THE TRANSACTION EXCLUSION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

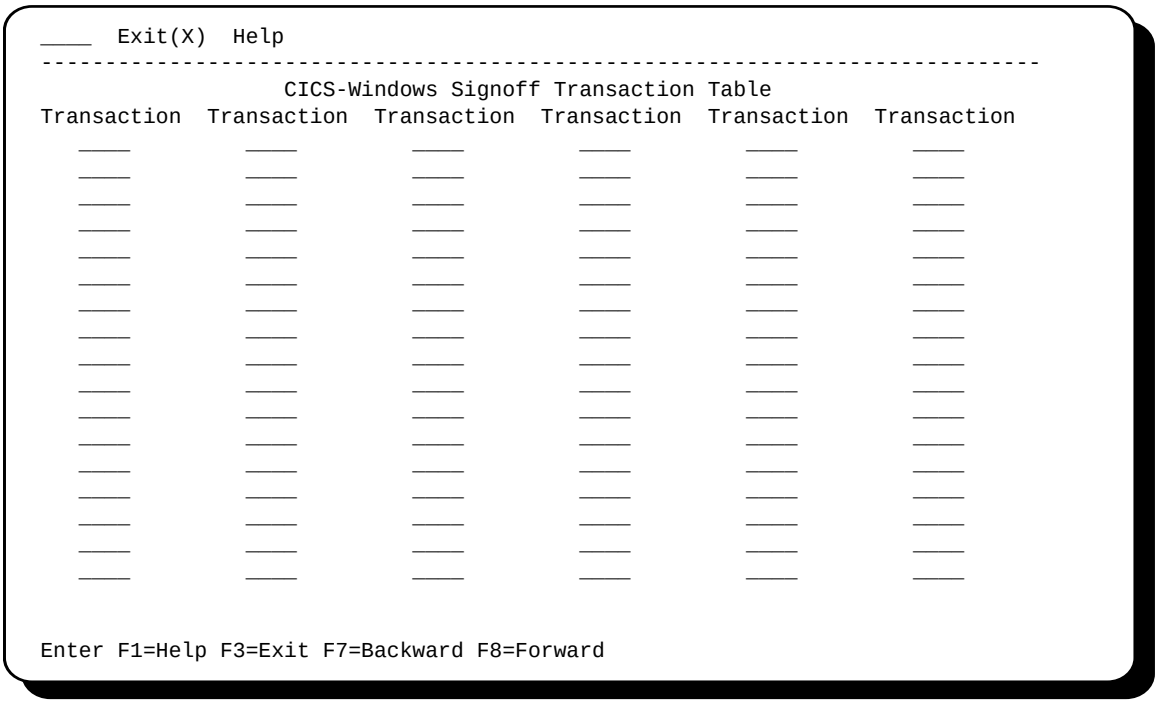
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE SIGNOFF TRANSACTION TABLE

The Sign-off Transaction table is used to define one or more transaction codes which designate a sign-off transaction.

If you are using something other than CSSF or CESF as a sign-off transaction code, or in DOS/VSE Interactive Interface if exiting the highest level menu is not the same as a sign-off, you will need to code all transaction codes as well.

To display the Sign-off Transaction table, from the Auxiliary Functions menu key a "8" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Signoff Transaction Table
Transaction Transaction Transaction Transaction Transaction Transaction
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE SIGNOFF TRANSACTION TABLE

TRANSACTION

The TRANSACTION operand is the only field of the Sign-off table. It is used to define one or more transaction codes which designate a sign-off transaction to CICS. If the only sign-off transaction code in use is CSSF or CESF or, when using the DOS/VSE Interactive Interface, if exiting the highest-level selection panel via PF3 (or other PF key) constitutes a sign-off, this statement need not be present. You need only specify transactions other than CSSF in this table. This statement is only necessary if you are using the REQUIRE WNDO,OFF AT LOGOFF and FORCE PURGE AT SIGNON AND SIGNOFF. See the discussion of *SESSION TERMINATION OPTIONS* in the section on *SPECIAL CONSIDERATIONS* for more details.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1) Code the full 4-character transaction code.

2) Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3) Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are not coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as sign-off transactions.

ENTERING COMMANDS AT THE SIGNOFF TRANSACTION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

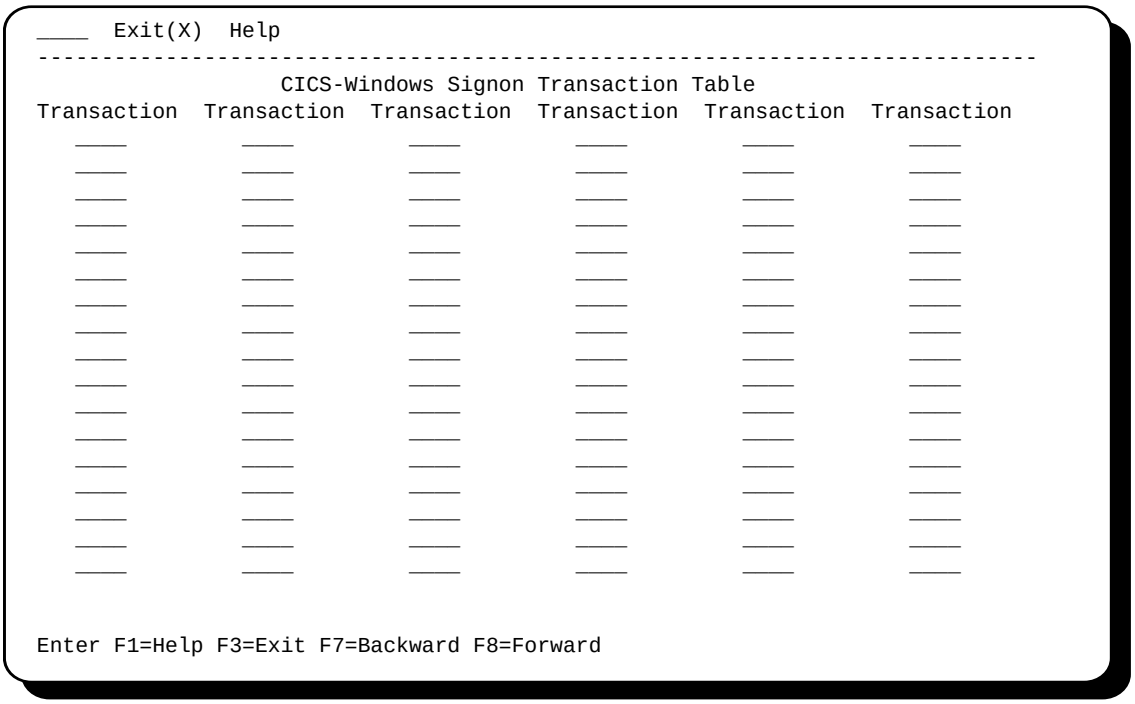
© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE SIGNON TRANSACTION TABLE

The Sign-on Transaction table is used to define one or more transaction codes which designate a sign-on transaction.

If you are using something other than CSSN or CESN as a sign-on transaction code, you will need to code all transaction codes as well.

To display the Sign-on Transaction table, from the Auxiliary Functions menu key a "9" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Signon Transaction Table
Transaction Transaction Transaction Transaction Transaction Transaction
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE SIGNON TRANSACTION TABLE

TRANSACTION

The TRANSACTION operand is the only field of the Sign-on table. It is used to define one or more transaction codes which designate a sign-off transaction to CICS. If the only sign-off transaction code in use is CSSF or CESN this statement need not be present. This statement is only necessary if you are using the FORCE PURGE AT SIGNON AND SIGNOFF option. See the discussion of *SESSION TERMINATION OPTIONS* in section 12 *SPECIAL CONSIDERATIONS* for more details.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the transaction code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are not coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as sign-off transactions.

ENTERING COMMANDS AT THE SIGNON TRANSACTION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

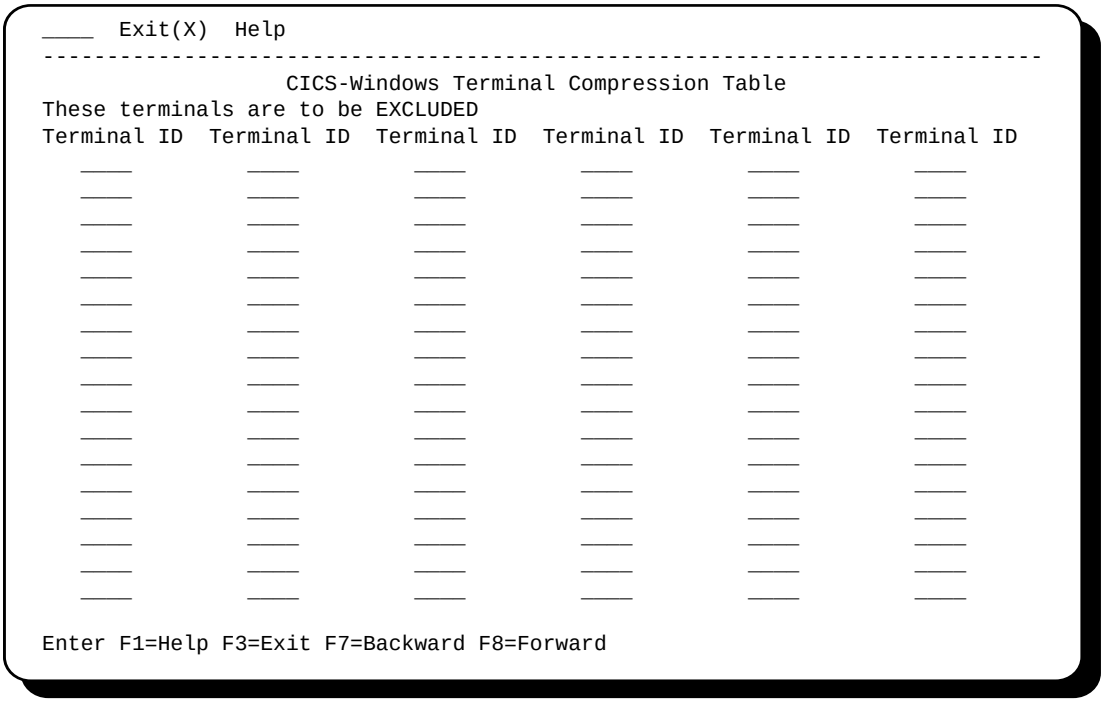
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE TERMINAL COMPRESSION TABLE

This table is used to define a table of terminal IDs which either are not to participate in the data compression feature of CICS-WINDOWS, even though the data compression feature is on, or are to be the only terminals that are eligible to participate in the data compression.

To display the Terminal Compression table, from the Auxiliary Functions menu key an "10" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Terminal Compression Table
These terminals are to be EXCLUDED
Terminal ID Terminal ID Terminal ID Terminal ID Terminal ID Terminal ID
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE TERMINAL COMPRESSION TABLE

TERMINAL ID This field is used to define one or more terminal IDs which are to be excluded from data compression or included as the only terminals whose datastreams are to be compressed.

Each terminal ID may be coded in one of three ways, any combination of which can be used together for different IDs:

1). Code the full 4-character terminal ID.

2). Code the first 1 to 3 characters followed by an asterisk (*). All terminals with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the ID. With this method, If all positions in the terminal ID match the corresponding positions in the TERMINAL ID operand where question marks are not coded, that terminal will be considered a match.

Do not code "*" or "????" (all wild-cards). This will specify all terminals.

THESE TRANSACTIONS ARE TO BE

This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED
All transactions on this table will not be allowed to toggle while using CICS-WINDOWS.

INCLUDED
Only the transactions appearing on this table will be allowed to toggle while using CICS-WINDOWS.

ENTERING COMMANDS AT THE TERMINAL COMPRESSION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

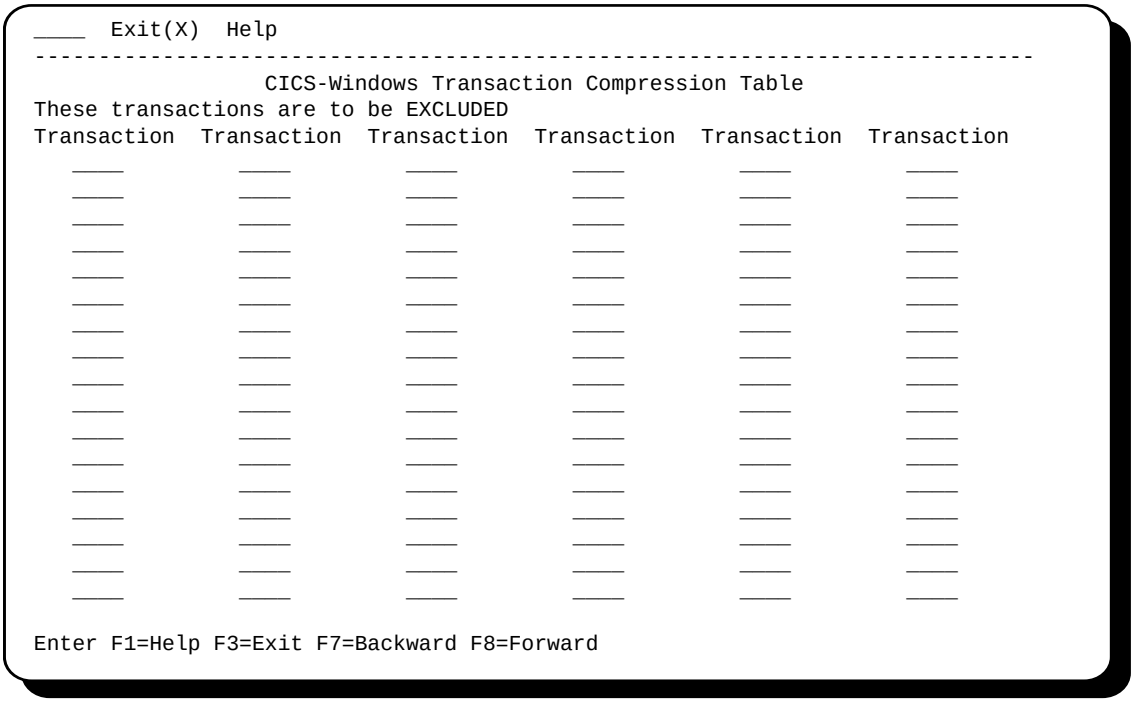
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE TRANSACTION COMPRESSION TABLE

The Transaction Compression table is used to define a table of transaction codes which are either not to participate in the data compression feature of CICS-WINDOWS, even though the data compression feature is on, or are to be the only transactions that will be eligible to participate in compression.

To display the Transaction Compression table, from the Auxiliary Functions menu key a "11" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Transaction Compression Table
These transactions are to be EXCLUDED
Transaction Transaction Transaction Transaction Transaction Transaction
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

FIELDS OF THE TRANSACTION COMPRESSION TABLE

THESE TRANSACTIONS ARE TO BE

This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED
: All Entries on this table will not be allowed to participate in compression.

INCLUDED
: Only the entries appearing on this table will be allowed to participate in compression.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TRANSACTION

This field is used to define one or more transaction codes which are to be excluded from data compression or included as the only transactions whose datastreams are to be compressed.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1) Code the full 4-character transaction code.

2) Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3) Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are not coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will specify all transactions.

ENTERING COMMANDS AT THE TRANSACTION COMPRESSION TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

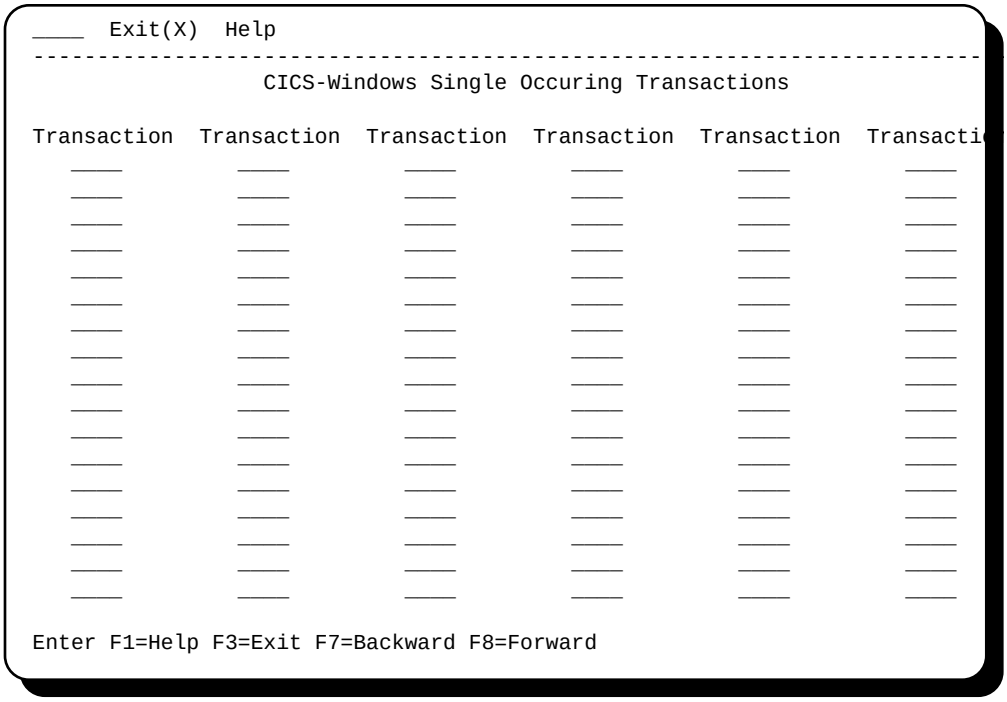
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE SINGLE OCCURRING TRANSACTIONS TABLE

This table is used to define one or more transaction codes which are “single-occurring,” meaning it can only be active in one virtual terminal per physical terminal at a time. This option should be used if you have certain packages or application systems which can not run in two sessions at once on the same physical terminal. There are very few transactions where this is a problem, but ADR’s VOLLIE system is one example of such a package. Running VOLLIE in multiple sessions on the same terminal can result in transaction ABENDS in the VOLLIE program. You should also consider using this feature if you are not using Pseudo terminal IDs to establish a unique terminal ID in each virtual terminal.

To display the Single Occurring Transactions table, from the Auxiliary Functions menu key a "12" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Single Occuring Transactions
Transaction Transaction Transaction Transaction Transaction Transacti
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE SINGLE OCCURRING TRANSACTIONS TABLE

TRANSACTION

This is the only operand of the Single Occurring Transactions table. It is used to define one or more transaction codes which are “single-occurring,” meaning it can only be active in one virtual terminal per physical terminal at a time.

Each transaction code may be coded in one of three ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the TRAN operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRAN operand where question marks are not coded, that transaction will be considered a match.

Do not code “*” or “????” (all wild-cards). This will allow a single transaction to operate in only one session..

ENTERING COMMANDS AT THE SINGLE OCCURRING TRANSACTIONS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

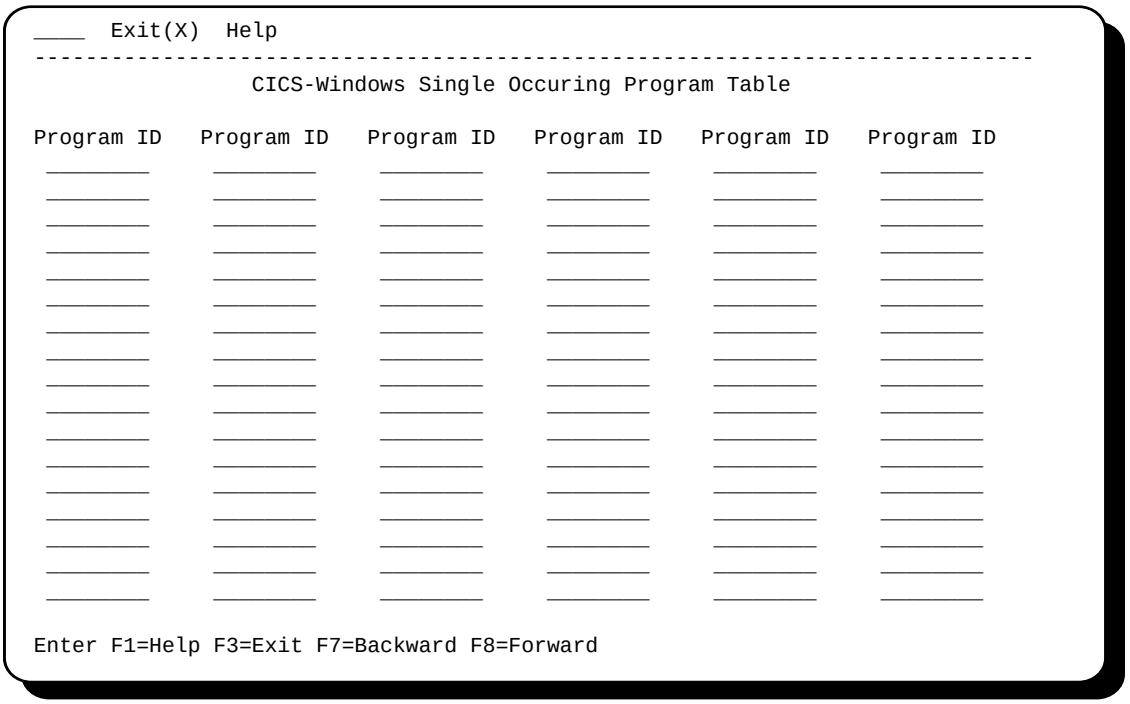
ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE SINGLE OCCURRING PROGRAMS TABLE

This table is used to define one or more program names which are “single-occurring,” meaning it can only be active in one virtual terminal per physical terminal at a time. This option should be used if you have certain packages or application systems which can not run in two sessions at once on the same physical terminal. You should also consider using this feature if you are not using Pseudo terminal IDs to establish a unique terminal ID in each virtual terminal.

To display the Single Occurring Programs table, from the Auxiliary Functions menu key a "13" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Single Occuring Program Table
Program ID Program ID Program ID Program ID Program ID Program ID
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE SINGLE OCCURRING PROGRAMS TABLE

PROGRAM ID

This is the only field of the Single Occurring Programs table. It is used to define one or more program names which are “single-occurring,” meaning it can only be active in one virtual terminal at a time.

Each program may be coded in one of the following ways, any combination of which can be used for different codes:

1). Code the full 8-character program name.

2). Code the first 1 to 7 characters followed by an asterisk (*). All programs with the same 1-7 characters (up to the asterisk), will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the name. With this method, If all positions in the program name match the corresponding positions in the PROGRAM ID operand where question marks are not coded, that program will be considered a match.

Do not code “*” or “????????” (all wild-cards). This will allow a single program to operate in only one session.

ENTERING COMMANDS AT THE SINGLE OCCURRING PROGRAMS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

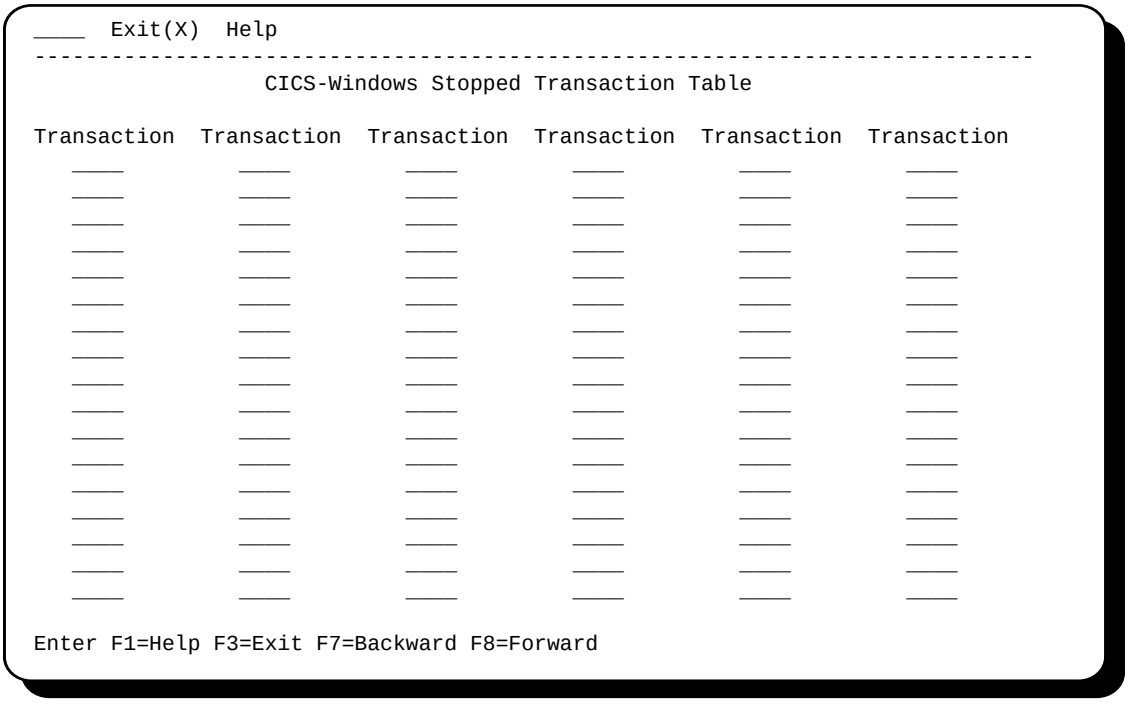
THE STOPPED TRANSACTIONS TABLE

This table is used to define one or more transaction codes which will not be allowed to execute if CICS-WINDOWS is active on the terminal. If one of the transaction codes in this table is entered, either in full-screen or window mode, the following message will display:

WNDO0022. xxxx CAN'T BE STARTED WITH CICS-WINDOWS

There are no known transactions that cannot run with CICS-WINDOWS. This table is provided as a "stop-gap" measure to prevent problems if one is encountered until Unicom technicians can resolve the conflict.

To display the Stopped Transactions table, from the Auxiliary Functions menu key a "14" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Stopped Transaction Table
Transaction Transaction Transaction Transaction Transaction Transaction
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE STOPPED TRANSACTIONS TABLE

TRANSACTION

This is the only field of the Stopped Transactions table. It is used to define one or more transaction codes that will not be allowed to execute if CICS-WINDOWS is active on the terminal.

Each transaction code may be coded in the following ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are not coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as non-executable transactions.

ENTERING COMMANDS AT THE STOPPED TRANSACTIONS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

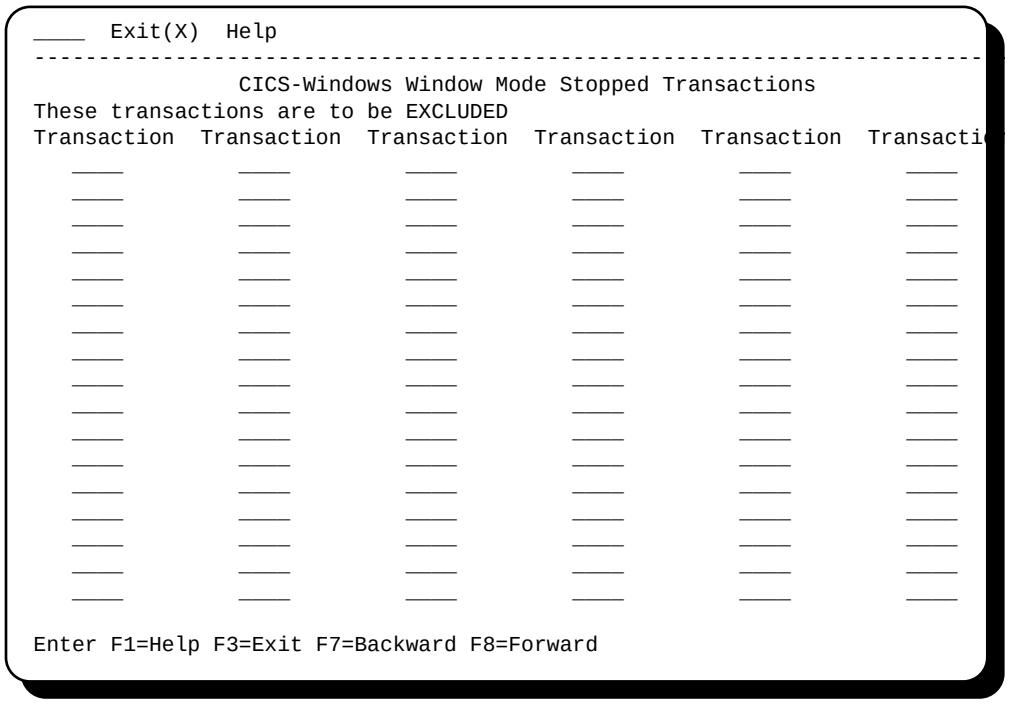
THE WINDOW MODE STOPPED TRANSACTIONS TABLE

This table is used to define one or more transaction codes which will not be allowed to execute in window mode. If a task attempts to execute a transaction in this table, the following message will display:

WNDO0011 TRANSACTION xxxx NOT ALLOWED IN WINDOW MODE

There are no known transactions that cannot run with CICS-WINDOWS. This table is provided as a "stop-gap" measure to prevent problems if one is encountered until Unicom technicians can resolve the conflict.

To display the Window Mode Stopped Transactions table, from the Auxiliary Functions menu key a "15" and press ENTER. This screen will appear generally as follows:


____ Exit(X) Help
CICS-Windows Window Mode Stopped Transactions
These transactions are to be EXCLUDED
Transaction Transaction Transaction Transaction Transaction Transacti
Enter F1=Help F3=Exit F7=Backward F8=Forward

You can change any value where the cursor will stop when the TAB key is pressed. The following discussion describes each field.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

FIELDS OF THE WINDOW MODE STOPPED TRANSACTIONS TABLE

THESE TRANSACTION ARE TO BE

This field determines if this table is to be an inclusion or an exclusion table. Valid entries are:

EXCLUDED
All entries on this table will not be allowed to operate in window mode.

INCLUDED
Only the entries appearing on this table will be allowed to operate in window mode.

TRANSACTION
This is the only field of the Window Mode Stopped Transaction table. It is used to define one or more transaction codes that will not be allowed to execute in window mode.

Each transaction code may be coded in the following ways, any combination of which can be used together for different codes:

1). Code the full 4-character transaction code.

2). Code the first 1 to 3 characters followed by an asterisk (*). All transactions with the same 1-3 characters (up to the asterisk) will be matched.

3). Define a generic definition using "wild-card" characters. Code the operand with question marks (?) in one or more positions of the code. With this method, If all positions in the transaction code match the corresponding positions in the TRANSACTION operand where question marks are not coded, that transaction will be considered a match.

Do not code "*" or "????" (all wild-cards). This will cause all transactions to be treated as non-executable transactions in window mode.

ENTERING COMMANDS AT THE WINDOW MODE STOPPED TRANSACTIONS TABLE

At the bottom of the screen is a list of available functions that may be invoked. You may invoke the desired option by pressing the associated PF key or by tabbing to the option and pressing ENTER.

ENTER	Apply any changes that you have made to the table.
PF1 HELP	Display a Help screen for information on the field that the cursor was in when the PF key was pressed.
PF3 EXIT	Return to the Auxiliary Functions menu.
PF7 BACKWARD	Browse backward to the previous page of table entries.
PF8 FORWARD	Browse forward to the next page of table entries.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 12. SPECIAL CONSIDERATIONS

PERFORMANCE TIPS

Generally speaking, CICS-WINDOWS adds so little overhead to the CICS system that performance is not much of a consideration. However, in certain situations, CPU overhead and/or storage overhead can impact overall system performance. In these cases there are certain customization options that can be avoided in order to effect less overhead, and others that can be activated in order to better control the environment.

Some situations where performance needs to be considered are as follows:

1) **Conversational Tasks.** If there are many conversational transactions in use, or if there are a few conversational tasks that are very high in storage utilization, you may want to place certain limits on the use of these transactions with CICS-WINDOWS. When a conversational task is toggled out-of, the dynamic storage in use by that task cannot be released. Thus, adding multiple session capability means that each operator can start more of these tasks, all of which will tie up resources.

   There are three ways to control this situation, any or all of which may be used:

   a) Use the Single-Occurring Program and/or Single-Occurring Transaction tables to limit each physical terminal to only one occurrence at a time of the transactions which are conversational.

   b) Use the Transaction Exclusion table to prevent an operator from toggling out of specified tasks. This will force the operator to terminate the task before any other session can be entered.

   c) Use the TIME OUT SELECTION TYPE option to cause CICS-WINDOWS to automatically purge any conversational task if the session containing that task is not entered for a specified interval of time.

2) **Extremely High Usage.** If you plan to allow 1500 to 2000 or more terminal operators to activate CICS-WINDOWS, CPU overhead may become a consideration. CICS-WINDOWS keeps an in-memory table called the "active user table" where an entry is kept for each active terminal. This table must be scanned at various points during operation in order to recognize an active terminal and test for certain activity to be handled by the WINDOWS program.

   You can reduce the amount of scan time and other overhead factors with the following options:

   a) Use READ BUFFER=YES. The READ BUFFER=NO option causes CICS-WINDOWS to get involved with every terminal output that occurs on every terminal. A scan must be performed to determine if this is a CICS-WINDOWS terminal and, if so, the terminal output is saved in order to reconstruct the terminal buffer when the toggle key is pressed.

      With READ BUFFER=YES, terminal output is not intercepted unless the terminal is in window mode and no datastream saving is performed. While it is true that a read-buffer

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

command takes longer to complete on a remote terminal, the overall savings in system performance may be worth it.

b) Use MODIFY TEMP STORAGE KEYS=NO in the User Options table and specify Pseudo Terminal IDs. MODIFY TEMP STORAGE KEYS should never be selected if pseudo IDs are in use. It has no effect when pseudo IDs are in use, but it causes a global CICS exit to be invoked on all temporary storage activity, and a scan of the active user table in order to determine that the terminal is using pseudo IDs.

c) Use MAIN storage rather than temporary. No I/O is required at all. The MAIN option will store all terminal images above the 16M line.

4) **Avoid Use of the Multiple Interactive Interface Feature.** The multiple interactive interface feature (VSE INTERACTIVE INTERFACE SUPPORT=YES) in DOS/VSE systems adds significant storage overhead to the system. It should not be used unless it is limited to a very few operators.

5) **Limit the use of windowing.** Windowing, having parts of several sessions displayed and active on the terminal screen at once, is a very useful option for many operators. However, it does increase the CPU overhead versus full-screen operation because on each invocation of a transaction, CICS-WINDOWS must first get control and then attach the user transaction. Likewise, the display from the application to the terminal must be re-windowed before presenting it on the screen.

It is often the case that MIS personnel want to use windowing, while end-users do not. This can be controlled by specifying separate profile for end-users with the NUMBER OF WINDOWS field set to zero.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INITIALIZING CICS-WINDOWS IN THE PLT

CICS-WINDOWS initializes during the first invocation of the WINDOWS phase. Usually this occurs the first time the WNDO transaction code is issued; however there are some benefits to initializing CICS-WINDOWS at PLT time:

- If AUTOSTART=YES is coded in any User Profile table, no terminals, operators or users can be auto-started until CICS-WINDOWS has been initialized.
- If your environment has multiple products that must be initialized in a certain order, you can ensure that the products are initialized correctly.

However you should read the following considerations before activating CICS-WINDOWS in the PLT:

- If you are installing CICS-WINDOWS for the first time or if this is a subsequent installation of a more current release level, you should not activate CICS-WINDOWS in the PLT until you are certain that the product has been installed correctly, as some error messages may not display on the system console.

  If this is a subsequent installation of a more current release level, you may encounter some difficulty migrating the customization options of your previous version of CICS-WINDOWS to the newer version.

If after reading the above, you do wish to initialize CICS-WINDOWS in the PLT, you will need to assemble and link-edit a system initialization Program List Table (or add WINDOWS to an existing table) as follows:

```
DFHPLT  TYPE=INITIAL,SUFFIX=xx
DFHPLT  TYPE=ENTRY,PROGRAM=WINDOWS
DFHPLT  TYPE=FINAL
```

NEWCOPY COMMANDS FOR WINDOWS PROGRAMS

Once activated, do not attempt to do a CSMT or CEMT NEW COPY command for the WINDOWS or WNDOTBL programs. Both programs are permanently "in-use" and the NEW COPY function may disable the program. This will result in APCT ABENDs for all transactions on that terminal, until the program is enabled.

In order to load a new version of either the WINDOWS or WNDOTBL modules you must use the WNDO,BACKOUT command, as described under *SPECIAL FUNCTION COMMANDS*, then perform the NEW COPY function.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PSEUDO TERMINAL ID CONSIDERATIONS

The use of Pseudo Terminal IDs is not required, but should be used in most CICS configurations. The Pseudo ID is used by CICS-WINDOWS to alter the terminal ID in the TCTTE (TCTTETI) each time the "toggle" key is pressed or each time a transaction is initiated from a window. The Pseudo ID will stay in the TCTTE until another virtual terminal is used or CICS-WINDOWS is ended on that terminal.

For terminal dependent applications (applications that base their logic on the terminal ID in any way) Pseudo IDs are necessary in order to run the same transaction in multiple virtual terminals on the same physical terminal.

The one exception to this is applications that use the terminal ID as part of a Temporary Storage key. This is a common practice and the use of pseudo IDs will resolve any conflicts. However, you can often operate without pseudo IDs by selecting the MODIFY TEMP STORAGE KEYS parameter of the User Options table, Temporary Storage keys containing terminal IDs are modified with the session number prior to writing or reading a Temporary Storage record. This technique prevents duplication of Temporary Storage keys, even though the terminal ID is the same in each session.

In general, you should use Pseudo IDs in any of the following situations:

- If applications are terminal dependent in some way other than by use of terminal ID with Temporary Storage.
- In an MRO/ISC environment where separation of terminal IDs in the remote region is required.
- Multiple Interactive Interface support for DOS/ VSE users.
- When time-initiated continual refresh transactions are in use.

If you have one of the situations listed above but do not want to use Pseudo IDs, you can avoid problems by defining a single-occurring program or single-occurring transaction table, as described in section 11 - *CUSTOMIZATION*. These tables prevent the simultaneous execution in multiple sessions of the same terminal of competing transactions.

When using Pseudo IDs, CICS terminal statistics are all accumulated for the real terminal ID only.

ACF2 users (prior to release 5.0) should not use Pseudo IDs. These versions of ACF2 require the operator to sign on in each virtual terminal if Pseudo IDs are in use. For release 5.0 and above, you can code the 'terminal identification' user exit in ACF2, using one of the interface routines for CICS-WINDOWS which obtains the real terminal ID and returns it to ACF2. For more information, please refer to *OBTAINING THE REAL TERMINAL ID FROM A PSEUDO ID* in section 15 - *USER EXITS AND PROGRAM INTERFACES.*

You might want to suppress or deactivate Pseudo IDs for certain transactions, usually editors, in order to get two versions of the transaction going on one physical terminal. Pansophic's O-W-L is a good example of this. If Pseudo IDs are set, O-W-L will not let you sign on to more than one virtual terminal. However, without Pseudo IDs, you can have multiple sessions of O-W-L running at a time.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If auto-initiated transactions which do continual timed displays are in use with CICS-WINDOWS, Pseudo Terminal IDs must be used. An auto-initiated task will only display in the virtual terminal where it was started if Pseudo Terminals are in effect. Without using Pseudo Terminals, when an auto-initiated task is started in one virtual terminal it will over-write the display in the next virtual terminal after pressing the "toggle" key.

For auto-initiated transactions to function properly in window mode, either Variable or Popup window mode must be in use.

CICS Emergency Restart

In CICS 1/7 and above, the TCTTE entries are logged to the restart dataset each time a terminal logs on. This includes the entries containing a pseudo terminal ID, if in use. Then, when an emergency restart is done, separate TCTTE entries are created for each pseudo terminal ID, which is not correct. If autoisnstall is used for TCT definitions, these pseudo entries are automatically deleted after a few seconds. While this condition does not really cause any problems with the operation of CICS, it means that excess storage is tied up for TCTTE entries that are never used.

ICCF REQUIREMENTS AND PECULIARITIES

DOS/VSE users with IBM's ICCF editor must be aware of the following conditions when using ICCF with CICS-WINDOWS.

- If you press any Toggle key after doing a SUBMIT LIBC, or $DA (any time initiated transaction in ICCF) before the completion message is displayed, you must use Pseudo Terminal IDs or the completion message will over-write the display in the next session. This will also cause an ICCF error.

- You must use Pseudo Terminal IDs if you are using the Forced Log-Off option in ICCF.

- You may have more than one session with ICCF active. DOS VSE (non SP or ESA) users may have the same or different ICCF Log-Ons in each session. VSE/SP and VSE/ESA users must have a different ICCF Log-on in each session.

- When running multiple ICCF sessions, you can not edit the same member in each session. (ICCF will prevent this).

- When running multiple ICCF sessions, you can not copy or move text from one session to the other. Each session has its own unique Scratch-Pad area.

- Because of the peculiarities of ICCF auto-initiated transactions, it is recommended that when running multiple ICCF sessions, you separate each ICCF session with a non-ICCF session. Thus, you could have four virtual terminals, ICCF in 1 and 3 and something else in 2 and 4.

  The only time you will encounter a problem by not doing this is in the case mentioned above, where you toggle out of a SUBMIT (or other time initiated task) before the completion message is displayed. When you do this with two ICCF sessions next to each other, the completion message will sometimes come back in the wrong session.

  [Note]: The above-mentioned peculiarity does not apply to VSE/SP and VSE/ESA users.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

- You must use Pseudo Terminal IDs to run multiple ICCF sessions.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USING THE CURSOR-SELECT KEY AS A HOT KEY

There is an additional key on most 3270 keyboards which can be used for any of the CICS-WINDOWS function keys (toggle, window, back-toggle, direct-session, window-switch or scrolling keys). It is the "cursor-select" key, usually labeled "cur sel" or "curs slct". It is not present on the older 3277 terminals, but is present on the 3278 and most later models.

To specify this key in the customization entries of the User Option Table, code CSEL as the key name for TOGGLE FORWARD, TOGGLE BACKWARD, WINDOW, SESSIONS DIRECT, WINDOW SWITCH OR SCROLL keys. To specify this key in response to the initial prompts for the toggle key or window key, simply press the cursor-select key, as you would any PF or PA key.

Use of the cursor-select key provides full functionality in most cases, but may take some special handling in some instances. The key is seldom used in application systems, which makes it a nice key to use as a "hot" key, since there is no conflict. However, you need to understand the hardware constraints of the key, and the special handling that CICS-WINDOWS does to allow its use in order to make a decision as to its usage in your environment. You may find that it is more trouble than it's worth.

The cursor-select key is provided as an alternative to the light-pen, for use in menu-selection type applications. IBM's intention was that you could program pen-detectable fields on the screen, position the cursor to the desired field and press the cursor-select key to make your selection. It simulates the light-pen in that it passes a special "aid" byte, along with the cursor position to the application program.

However, if the cursor is not positioned in a pen-detectable field where the first position of that field is either a blank (x'40') or a null (x'00'), the key will not cause an interrupt and will not function at all. You get a keyboard-inhibit indicator and must press RESET to free the keyboard and continue.

In order to allow the cursor-select key to be used, CICS-WINDOWS does the following, if the cursor-select key is used as any designated function key:

- Every displayable field of every output display on this terminal will have the "pen-detectable" attribute bit set on.
- If no fields are present in the output display (no attributes are found), a pen-detectable attribute is added to the end of the datastream.
- If the cursor is not explicitly positioned in the output display, the cursor will be positioned to the pen-detectable attribute at the end of the datastream.
- When the CLEAR key is pressed, a datastream consisting of one unprotected, pen-detectable field is written back to the screen.

[Note]: These modifications are done only for the terminal where the cursor-select key is designated as a CICS-WINDOWS function key and not for any terminal where it is not in use.

By performing these modifications to the output datastreams, you will find that in many cases, the cursor will be positioned to a blank, pen-detectable field on the screen. Pressing the cursor-select key at that point will cause an interrupt and the designated function will be performed.

If the cursor is not positioned to a blank field, if the first position of the field wherein it is positioned is blank, it will still work.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If the first position of the field where the cursor is positioned is non-blank, you can often move the cursor to a blank spot on the screen and press the cursor-select key and it will work. The same rules apply, that is, the first position of that field must be blank. It does not matter whether the field is protected or unprotected.

If none of the above seems to work, you can tab to an unprotected field on the screen and erase the first position of that field (or erase the whole field). You will now be able to use the cursor-select key. However, when you toggle back into this session, the data that you erased is still erased. CICS-WINDOWS has no way of restoring the field.

Also, you can always clear the screen, which causes the entire screen to be set to unprotected, pen-detectable, thereby allowing the cursor-select key to work.

As previously stated, you may find this more trouble than it is worth. However, depending on the application screens in your environment, you may find that it works quite well, thereby giving you the availability of a "hot" key which does not conflict with your user applications.

[Note]: You should not attempt to use the cursor-select key if you have applications that depend on pen-detectable fields, using either the light-pen or the cursor-select key for application logic.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SESSION TERMINATION OPTIONS

Depending on your security requirements, the choice of the session termination options in the User Option Table (options REQUIRE WNDO,OFF AT LOGOFF, REQUIRE CLEAR SESSIONS BEFORE OFF, REQUIRE TRANSACTION END BEFORE OFF, and FORCE PURGE AT SIGNON AND SIGNOFF should be carefully considered). Each option offers a slightly different variation to the action that CICS-WINDOWS performs at sign-off, sign-on or WINDOWS termination time.

You must consider four situations:

1). What happens when a WNDO,OFF command is issued?

2). What happens when a CICS sign-off transaction is issued?

3). What happens when another CICS sign-on transaction is issued after previously signing on and activating CICS-WINDOWS?

4). What happens, in VTAM systems, when, for any reason, the terminal is dropped from CICS and returns to VTAM?

Referring to these four questions, the following discussion describes the action taken by CICS-WINDOWS for each option, as well as the absence of all four options:

[Note]: There is another alternative to handling session termination which involves using one of the program interface routines. This involves calling CICS-WINDOWS with a PURGE command when an operator signs off and also when an operator signs on. Two sample programs, WNDOCSSN and WNDOCSSF are present in the first file on the installation tape which do these functions. See *SAMPLE PROGRAMS ON THE INSTALLATION TAPE* in section 15 - *USER EXITS AND PROGRAM INTERFACES* for more information.

If no options are coded (default action).

Question 1. What happens at WNDO,OFF?

If there are no conversational tasks active in any session, CICS-WINDOWS is terminated. Otherwise, the following message will display:

WNDO1409. ACTIVE TASK xxxx ON LOGICAL TERMINAL n MUST BE ENDED.

The conversational transaction must be ended, then the WNDO,OFF command must be re-issued.

Question 2. What happens at CICS sign-off?

The sign-off is allowed to complete, then CICS-WINDOWS starts a task at the terminal to issue the message:

WNDO0004. DO YOU WANT TO TERMINATE CICS-WINDOWS ?

The operator replies with 'Y' or 'N'. Replying 'Y' causes normal WNDO,OFF logic to be taken as described in question 1, above. Replying 'N' allows the operator to leave this CICS system and go to another system, then return to this CICS, press the toggle key and have the session restored.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

[Note]: Since this message appears after sign-off is complete, you must not code security on the WNDO transaction for this to work correctly.

Question 3. What happens with a subsequent CICS sign-on?

No action is taken, the operator can sign-on again without interruption.

Question 4. What happens if the terminal is dropped?

No action is taken. The operator may sign on again or simply re-enter CICS after re-acquiring the terminal. Note, however, that this can sometimes be a dangerous situation, depending on the security system in use. If CICS-WINDOWS is not terminated, previously saved addresses that are no longer in use can be restored, resulting in storage violations.

If the Require Transaction End Before Off option is coded.

Question 1. What happens at WNDO,OFF?

If there are no conversational or Pseudo-conversational tasks active in any session, CICS-WINDOWS is terminated. Otherwise, the message

WNDO1409. ACTIVE TASK xxxx ON LOGICAL TERMINAL n MUST BE ENDED.

will display. The transaction must be ended, then the WNDO,OFF command re-issued.

CICS-WINDOWS recognizes a pseudo-conversational task by the presence of a return transaction code (next-tran) in the session.

Question 2. What happens at CICS sign-off?

The sign-off is allowed to complete, then CICS-WINDOWS starts a task at the terminal to issue the message:

WNDO0004. DO YOU WANT TO TERMINATE CICS-WINDOWS ?

The operator replies with 'Y' or 'N'. Replying 'Y' causes normal WNDO,OFF logic to be taken as described in question 1, above. Replying 'N' allows the operator to leave this CICS system and go to another system, then return to this CICS, press the toggle key and have the session restored.

[Note]: Since message 0004 appears after sign-off is complete, you must not code security on the WNDO transaction for this to work correctly.

Question 3. What happens with a subsequent CICS sign-on?

No action is taken, the operator can sign-on differently or the same in each session.

Question 4. What happens if the terminal is dropped?

No action is taken. The operator may sign on again or simply re-enter CICS after re-acquiring the terminal. Note, however, that this can sometimes be a dangerous situation, depending on the security system in use. If CICS-WINDOWS is not terminated, previously saved addresses that are no longer in use can be restored, resulting in storage violations.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If the Require Clear Sessions Before Off option is coded.

This option works exactly like REQUIRE TRANSACTION END BEFORE OFF, in that it prevents CICS-WINDOWS termination if there are any active sessions. However, it is more stringent that REQUIRE TRANSACTION END BEFORE OFF in that it requires that every session screen except the one where the WNDO,OFF command is being issued be absolutely clear.

Usually this is unnecessary, and makes it quite difficult to terminate CICS-WINDOWS. Therefore it is not recommended. It is provided for those users who have non-conversational tasks, not using RETURN TRANSID, that must be ended before terminating WINDOWS.

If the Require WNDO,OFF At Logoff option is coded.

Question 1. What happens at WNDO,OFF?

Normal WNDO,OFF logic is used, as described above. If either REQUIRE TRANSACTION END BEFORE OFF or REQUIRE CLEAR SESSIONS BEFORE OFF are coded, their logic will be invoked.

Question 2. What happens at CICS sign-off?

If CICS-WINDOWS is active on the terminal, the sign-off is not allowed to complete. The following message is displayed:

WNDO0009. CICS-WINDOWS MUST BE TERMINATED TO LOG-OFF

The operator must issue a WNDO,OFF command to terminate CICS-WINDOWS, which will take normal logic as described above, then re-issue the sign-off transaction.

[Note]: Since message 0009 appears before sign-off is complete, you can code security on the WNDO transaction with this option.

Question 3. What happens with a subsequent CICS sign-on?

When CSSN, or your sign-on transaction is entered and CICS-WINDOWS is active on the terminal, it is assumed that it is a second sign-on (REQUIRE WNDO,OFF AT LOGOFF implies that you sign-on before performing a WNDO,ON command). With REQUIRE WNDO,OFF AT LOGOFF, the subsequent sign-on is treated like a sign-off and message 0009 is displayed as described above. Transactions other than CSSN which perform a sign-on should be defined in the Sign-on Transaction table.

Question 4. What happens if the terminal is dropped?

No action is taken. The operator may sign on again or simply re-enter CICS after re-acquiring the terminal. If the operator signs on, it will be treated as described in question 3, above. Note, however, that this can sometimes be a dangerous situation, depending on the security system in use. If CICS-WINDOWS is not terminated, previously saved addresses that are no longer in use can be restored, resulting in storage violations.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If the Force Purge at Signon and Signoff option is coded.

Question 1. What happens at WNDO,OFF?

Normal WNDO,OFF logic is used, as described above. If REQUIRE WNDO,OFF AT LOGOFF, REQUIRE TRANSACTION END BEFORE OFF or REQUIRE CLEAR SESSIONS BEFORE OFF are coded, their logic will be invoked.

Question 2. What happens at CICS sign-off?

If CICS-WINDOWS is active on the terminal, the terminal is purged from CICS-WINDOWS, then the sign-off is allowed to complete.

Question 3. What happens with a subsequent CICS sign-on?

CICS-WINDOWS is automatically terminated with a PURGE command before a new sign-on is accepted. This means that any conversational transactions that are active on the terminal are abnormally ended with an AKCS abend.

This option effectively prevents multiple sign-ons at a terminal, plus insuring that CICS-WINDOWS is off when a user tries to sign-on again.

[Note]: If your sign-on transaction is something other than 'CSSN' or 'CESN', it should be coded in the Sign-on/Sign-off Transaction table.

Question 4. What happens if the terminal is dropped?

No action is taken at the time the terminal is dropped, since CICS-WINDOWS usually can not recognize when this occurs. However, assuming that the operator must sign-on in order to re-acquire the terminal, it will be treated as described in question 3, above. This option effectively resolves the danger of storage violations described previously.

[Notes]:

1). If you are using a security package, you should always use FORCE PURGE AT SIGNON AND SIGNOFF. Most security packages release their control blocks at sign-off. Therefore leaving CICS-WINDOWS active after signing off can cause a bad control block address to be restored, causing a possible system abend or storage violations.

Security packages RACF, ACF2, TOP SECRET and SURVEILLANCE are known to require these options.

2). Even if you don't have a security package, omitting REQUIRE WNDO,OFF AT LOGOFF or PURGE AT SIGNON AND SIGNOFF involves a security risk, since it allows a new operator to toggle into a session transaction which they might not be authorized to use.

3). In general, REQUIRE TRANSACTION END BEFORE OFF and REQUIRE CLEAR SESSIONS BEFORE OFF need never be used. It causes no problems to terminate a session when a pseudo-conversational transaction is active in that session, since the transaction is at a logical sync point anyway. It's the same situation as if CICS came down while a pseudo-conversational transaction was awaiting input.

However, if database integrity could be disturbed by ending a pseudo-conversational transaction without going through normal completion, you should use one of these options.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Using the Inactivity Time-Out Feature

The inactivity time-out feature of CICS-WINDOWS is used to ‘lock’ a terminal after a specified period of no input from the operator. The selection of this feature and the specification of the time-out interval are specified in the User Options Table of the customization processor.

When a time-out occurs, CICS-WINDOWS performs the following function:

1) If the task which produced the current display is not conversational, all session information is saved and a screen is displayed requesting the operator to enter their name and password.

2) If the current session task is conversational, it is not disturbed. However, when the next input is received from the terminal, the operator sign-on screen mentioned above will then display.

3) No input will be accepted from the terminal until the operator enters a valid name and password. That is, all toggle keys, window keys and any other CICS-WINDOWS function will be ignored. If the Clear key is pressed, the sign-on screen will redisplay. Enter, PF3, or any other function key not assigned to CICS-WINDOWS will result only in a redisplay of the sign-on screen.

4) When the operator enters a name and password, CICS-WINDOWS calls the DFHXSP global security interface of CICS to validate the incoming name and password. This interface handles normal CICS security calls and is documented in the CICS Customization Guide.

5) If the name and password are accepted, CICS-WINDOWS will redisplay the last screen that was displayed prior to the time-out and normal function may be resumed. CICS-WINDOWS is still active on the terminal, and all inactive sessions may be re-entered by toggling to them, if desired.

6) If the name and password are invalid, the sign-on screen will continue to display until a valid sign-on is entered.

If you are using standard CICS security for any release of CICS, nothing is required to make this feature work properly beyond activating it on the customization screen. For non-standard security packages, you must ensure that your security system conforms to the coding conventions documented in the Customization Guide for the DFHXSP security interface.

[Note]. CICS-WINDOWS must be active on a terminal for the inactivity time-out to occur.

WLOC Transaction

The transaction code WLOC may be used to directly invoke the terminal ‘lock’ without the occurrence of a time-out. Simply enter WLOC from a clear screen. The CICS-WINDOWS logo display requesting name and password will appear and the terminal is locked at the point until the sign-on information is entered.

Optionally, WLOC can be assigned to a PF key using the TASKREQ field of the PCT entry for the WLOC transaction. If this is done, the operator can press the designated PF key as long as the task in progress is not conversational. In other words, a clear screen is not required to invoke the lock function.

When invoked directly with WLOC, the current screen and session information is saved in the same manner as when a time-out occurs, and restored when the name and password are entered.

[Note]. The WLOC transaction can be renamed to any transaction code desired.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The transaction code WLSC is used to control the inactivity time-out feature. It must be defined to CICS as described in *Installation* and point to program WNDOLOCK.

Under normal operation, the WLSC transaction need never be entered. CICS-WINDOWS will invoke it automatically when it is first activated with the *Lock Inactive Terminals* option selected. However, you may want to use this transaction from time to time to temporarily disable or inactivate the time-out feature.

To use the WLSC transaction, enter the following from a clear screen:

WLSC,*command*,[TERM=*xxxx*]

Where *command* is one of the following:

1) STOP — Stop inactivity polling (Deactivate the time-out feature.) Applies globally, do not code TERM=*xxxx*.

2) START — Start inactivity polling (Activate the time-out feature.) Applies globally, do not code TERM=*xxxx*.

3) DISABLE — Turn off the inactivity time-out for a specific terminal. TERM=*xxxx* is required.

4) ENABLE — Turn on the inactivity time-out for a specific terminal. TERM=*xxxx* is required.

[Note]. If the inactivity time-out is stopped with a WLSC,STOP command, the only way to start it again is with a WLSC,START command or bringing CICS down and back up. Use of the WNDO,BACKOUT command will not override a WLSC,STOP.

[Note]. TERM=ALL may be specified on the DISABLE and ENABLE commands to apply to all terminals, if desired.

[Note]. The WLSC transaction code can not be renamed to any other transaction code.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 13. UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS

The following discussion concerns the special considerations and facilities provided for handling different CICS environments and situations. In particular, the MRO/ISC environment and the VSE Interactive Interface are discussed.

RUNNING CICS-WINDOWS WITH MRO/ISC

If the Multiple Region Operation (MRO) or Inter-System Communication (ISC) feature of CICS is in use, the following considerations will apply:

- All CICS-WINDOWS programs must be loaded in the Terminal Owning Region, not in the remote regions. If there are multiple Terminal Owning Regions, they may be loaded in each one.

- If Pseudo Terminals ID's are not specified (all virtual terminals have the same terminal ID), CICS-WINDOWS will work and you can effectively toggle across regions through MRO, using either the CRTE transaction or remote transactions in the PCT. However, the following conditions can occur:

1) When a remote transaction is started from a WINDOWS virtual terminal that is conversational (issues a terminal read), the remote TCT entry is considered active by CICS until that transaction is ended.

   If you then press the "toggle" key and move to another session, then start another transaction routed to the same remote region, you will receive CICS Abend ATZW, which says that an attempt was made to attach a task to a remote terminal that was already running a task. The Abend will not effect anything going on in the other sessions and if you toggle back to the conversational transaction and end it, you may then start another remote task to that system.

2) When a remote pseudo-conversational program is started which uses a COMAREA, then the operator toggles to another virtual terminal and starts another transaction in the same region which also uses a COMAREA, the second transaction will, on some occasions, obtain the COMAREA that was established by the first transaction. This can cause unpredictable results.

- When Pseudo Terminal IDs are in use, for CICS 1.7 and above, you should use the "shippable" TCT option of CICS. This option allows the remote CICS region to accept whatever Terminal ID is sent from the Terminal Owning Region. For version 3 of CICS, shippable TCT is required (WNDOMRO is no longer available).

- For all CICS releases prior to 1.7 or if shippable TCTs are not in use, if Pseudo Terminal IDs are in use, something must be done in the remote regions in order to recognize the pseudo terminal ID and correlate it with the real sending terminal ID. There are two approaches to this problem:

1) Defining dummy remote TCT entries.
2) Using the WNDOMRO program.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DEFINING DUMMY REMOTE TCT ENTRIES

Generate dummy terminal entries in the remote CICS Terminal Control Table (TCT), with terminal ID's that match the Pseudo ID's in the sending region. In order to control the specification of Pseudo ID's, code the Auto-Start table, pre-defining all Pseudo ID's that will be in use in the sending region, thereby eliminating the possibility that a terminal operator will specify an ID (at WINDOWS initiation) that has not been defined in the remote region.

If this approach is used, you need to define the dummy terminal entries in the remote system's TCT only, not in the sending region (terminal-owning region).

[Note]: If PSEUDO IDS=YES is coded in the User Option Table, the generated Pseudo IDs contain lower-case characters (see the discussion of *GENERATED PSEUDO TERMINAL IDS* in section 11 - *CUSTOMIZATION*). Unless you have an editor which will let you enter lower-case, it is difficult to code these values in the remote TCT. We suggest, therefore that you explicitly define all terminals in the remote regions using Auto-Init table statements.

This approach provides total flexibility for all CICS-WINDOWS functions. Each dummy remote terminal entry requires 20 bytes of storage in the remote TCT, so storage overhead should not be a major consideration.

However, if your installation contains a large number of terminals, and coding the Dummy TCT entries is a problem, you may want to use the second approach, below.

USING THE WNDOMRO PROGRAM

The installation tape includes the WNDOMRO program as one of the link-edited object modules. Note that WNDOMRO cannot be used in version 3 of CICS; you must use shippable TCTs.

The WNDOMRO program must be link-edited and defined in the remote regions, along with the Auto-Start table, as described below. This program provides a facility in the remote system CICS to recognize a Pseudo ID and use it in building the surrogate terminal entry, thereby eliminating the need for dummy TCT entries.

Use of the WNDOMRO program in the remote regions will provide full flexibility and support for all CICS-WINDOWS with no restrictions:

To install and activate the WNDOMRO program, perform the following steps.

a). The WNDOMRO program is link-edited along with the other programs during installation. If the remote regions use the same Core-Image or Load library as the terminal owning regions, nothing more need be done. If different libraries are in use for the remote regions, copy or move the WNDOMRO program to the desired library. This could be performed using LIBR in DOS/VSE or TSO in MVS.

b). Define the VSAM file, WNDOFIL, which contains all customization statements, as a remote file available to the AOR. WNDOMRO must have access to this file.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

[Note]: You must define your Auto-Init table statements using either the TERM keyword or using USER=*. Individual or group Auto-Init table statements that define operators with the USER keyword may be used in the Terminal Owning Region, but will be ignored by the WNDOMRO program in the remote regions. For total flexibility, we recommend that you have a TERM=* or USER=* statement as the last (or only) Auto-Init table statement.

c). Define the following PPT entries in the remote region only:

DFHPPT TYPE=ENTRY,PROGRAM=WNDOMRO,RES=YES

DFHPPT TYPE=ENTRY,PROGRAM=WNDOTBL,RES=NO
(only needed if PSGNXIT is coded)

DFHPPT TYPE=ENTRY,PROGRAM=WNDOAUXL,RES=NO

DFHPPT TYPE=ENTRY,PROGRAM=DFHPLTxx
(xx=PLT SUFFIX, below)

d). Assemble and Link-Edit a system initialization Program List Table (or add WNDOMRO to an existing table) as follows:

```
DFHPLT  TYPE=INITIAL, SUFFIX=xx
DFHPLT  TYPE=ENTRY,PROGRAM=WNDOMRO
DFHPLT  TYPE=FINAL
```

e). Add the following System Initialization override parameter to your CICS startup JCL for the remote region(s):

PLTPI=xx

where xx is the DFHPLT suffix. After completion of these steps, you should receive the following message during the CICS startup in the remote region:

WMR0100. WNDOMRO SUCCESSFULLY ACTIVATED

You may now activate CICS-WINDOWS in the Terminal Owning Region, using Pseudo Terminal IDs. Full separation of terminal IDs is provided in the remote region and you can have any mixture of transaction types (conversational, pseudo-conversational and non-conversational) running on one terminal.

MULTIPLE TERMINAL OWNING REGIONS

If there is only one Terminal Owning Region and one or more application (remote) regions, the distribution of programs when using WNDOMRO is as follows:

- All CICS-WINDOWS programs must be installed in the Terminal Owning Region.
- WNDOMRO and WNDOAUXL must be installed in each remote region. If the PSGNXIT operand is coded, then WNDOTBL must be installed as well.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If there are multiple Terminal Owning Regions where a given region can be both Terminal Owning and remote, you must install all CICS-WINDOWS programs in all regions. The rules to follow are:

- If a region has real terminals defined in its TCT (Terminal Owning Region), that region must have the WINDOWS, WNDOMAIN, WNDOENAB, WNDOINIT and optionally WNDOTBL programs defined locally in its PPT and PCT.

- If a region has remote terminals defined in its TCT (either exclusively or in addition to real terminals), that region must have the WNDOMRO, WNDOAUXL, and optionally WNDOTBL programs defined in its PPT and PCT.

- The VSAM control file, WNDOFIL, must be available to all regions.

[Note]: None of the above considerations apply if shippable TCTs are in use. In that case, install all CICS-WINDOWS programs in every TOR. Either use separate VSAM files or define the same file as accessible from multiple TORs, and nothing more is required. Do not define WNDOMRO anywhere.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

RUNNING WINDOWS WITH THE DOS VSE INTERACTIVE INTERFACE

For DOS VSE users where the Interactive Interface is in use, CICS-WINDOWS provides three methods of operation. The decision is whether to have an Interactive Interface selection panel (menu) in each virtual terminal or only one and how to control the operator sign-on procedure.

The three methods of operation are as follows:

1) Run with a selection panel in the first virtual terminal only, using only one operator sign-on.

   With this method, the selection panel can be used to initiate transactions in the first session only. In all other sessions, transactions must be started using transaction code entry. When you are in the other sessions and try to initiate the Interactive Interface, you receive a message from CICS-WINDOWS to the effect that multiple interactive interface menus are not allowed.

   [Note]: This method does not require that Pseudo Terminal IDs be in use. Each virtual terminal may have the same ID as the real terminal or it may have a unique ID.

2) Run with the same selection panel in all virtual terminals, using only one operator sign-on.

   With this method, the selection panel that was activated when you first signed-on to CICS is automatically started in each virtual terminal the first time you toggle into that terminal. Each time you press PF3 to return to the selection panel, the selection panel is restored, no matter what virtual terminal you are in.

   Full support for all options of the selection panel is provided with the following restriction:

   If you are using the system supplied selection panel, you can not select an option which activates ICCF from more than one virtual terminal at a time. You can have multiple versions of ICCF active, however, by using PF6 to exit the selection panel and starting ICCF from a transaction code, as long as you use a different ICCF log-on.

   If you exit a selection panel using PF3, which signs you off from CICS, you can not sign back on using the same CICS sign-on. You must either use a different sign-on, sign-off of all sessions, or do a WNDO,OFF.

   [Note]: This method requires that Pseudo Terminal IDs be in use. Each virtual terminal must have a unique ID.

3) Run with the same or different selection panel in each virtual terminal using a different CICS sign-on in each session.

   With this method, the first time that you toggle into an unused virtual terminal, you are presented with the CICS sign-on screen. You must sign-on with a unique operator name and password in each session. The selection panel associated with this sign-on will be started in this virtual terminal. This can be the same panel or a different panel from the one started when you first signed-on to CICS.

   All options of every selection panel are supported in each virtual terminal, including multiple sessions of ICCF started from the system supplied selection panel.

   If you exit a selection panel using PF3, which signs you off from CICS, you can sign back on using the same CICS sign-on that was previously in use in that virtual terminal (or with a sign-on that is unique from the remaining sessions).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If you do a WNDO,OFF command, you will automatically be signed-off from all virtual terminals except the one you are currently in.

[Note]: This method requires that Pseudo Terminal IDs be in use. Each virtual terminal must have a unique ID.

INSTALLING THE MULTIPLE INTERACTIVE INTERFACE FEATURE

You can choose one method of operation with the Interactive Interface that all terminals will use, or you can mix the three methods by terminal, if desired. The User Profile Table is used to specify the type of operation desired. In addition, if method 2 or method 3 is to be used, you must install the WNDOVSP program in the same CICS partition with the WINDOWS program. If you do nothing other than install the basic package, method 1 (selection panel in virtual terminal 1 only) is the default.

CHOOSING THE METHOD OF OPERATION

You must code the VSE INTERFACE SUPPORT operand (on the User Profile table) as YES, in order to enable the use of the Interactive Interface. Code one of the following operands on each User Profile table:

- VSE INTERACTIVE INTERFACE=NO This specifies that no multiple Interactive Interface sessions be allowed for this terminal. Method 1, described above, will be in force for all terminals using this profile.

- VSE INTERACTIVE INTERFACE=YES This specifies that Method 2, described above, will be in force for all terminals using this profile. Multiple selection panels are supported using only one operator sign-on.

- VSE INTERACTIVE INTERFACE=SIGNON This specifies that Method 3, described above, will be in force for all terminals using this profile. Multiple selection panels are supported, requiring a different operator sign-on in each session.

  [Note]: You must define Pseudo Terminal IDs in the Auto-Init table entries if the VSE INTERFACE SUPPORT value for that entry is YES or SIGNON.

INSTALLING THE WNDOVSP PROGRAM

The WNDOVSP program must be installed if you specify VSE INTERACTIVE INTERFACE as 'YES' or 'SIGNON' on any User Profile Table. There must be a PPT entry for WNDOVSP in the same CICS partition where the WINDOWS program is installed and the WNDOVSP program must be defined in a start-up DFHPLT table.

To install and activate the WNDOVSP program, perform the following steps.

a) The WNDOVSP program is link-edited along with the other programs during installation.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

b) Define the following PPT entries:

DFHPPT TYPE=ENTRY,PROGRAM=WNDOVSP,RES=NO

DFHPPT TYPE=ENTRY,PROGRAM=DFHPLTxx
(xx=PLT SUFFIX, below)

c) Assemble and Link-Edit a system initialization Program List Table (or add WNDOVSP to an existing table) as follows:

DFHPLT TYPE=INITIAL, SUFFIX=xx
DFHPLT TYPE=ENTRY,PROGRAM=WINDOWS
DFHPLT TYPE=FINAL

d) Add the following System Initialization override parameter to your CICS startup JCL for the remote region(s):

PLTPI=xx

where xx is the DFHPLT suffix.

After completion of these steps, you should receive the following message during the CICS startup:

WNDOVSP SUCCESSFULLY ACTIVATED

ACTIVATING CICS-WINDOWS FROM A SELECTION PANEL

There are two ways to activate CICS-WINDOWS when using Interactive Interface selection panels:

1) Exit the selection panel using PF6 and start CICS-WINDOWS the normal way with a WNDO,ON command. After activating it, when you press PF3 you will return to the selection panel. When you press the Toggle key to move to the next virtual terminal, either a blank screen, the selection panel or the sign-on screen (depending on the VSE INTERFACE SUPPORT option chosen for this terminal) will be displayed.

2) Put CICS-WINDOWS as a selection on a selection panel. Using the PANEL TAILORING option of the system supplied selection panel, you can add CICS-WINDOWS as one of the selections to any selection panel.

To do this, first define an Application Profile which will start CICS-WINDOWS. You must use the ATTACH TRANSACTION WITH DATA option, the transaction code is "WNDO", and the data to be passed to the program is ',ON' (in uppercase). Then add a sequence number to a selection panel and provide the name of the application profile for that sequence number.

Upon completion you will be able to select CICS-WINDOWS from the selection panel and it will be activated. Likewise, you could have the WNDO,OFF command be a selection panel option, if desired.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Many of the options of the system supplied selection panel for VSE invoke ICCF. As previously mentioned, you can not enter ICCF from two selection panels of the same physical terminal unless you are using two different operator sign-ons. This is because VSE/SP and VSE/ESA ICCF will not allow two sessions to be started using the same log-on, as is possible on non-SP systems.

You can, however, start one ICCF session from a selection panel, toggle, exit the selection panel with PF6 and start another ICCF session by entering the ICCF transaction code and logging on with a different Operator ID. When you do this, you must log off from ICCF and press PF3 in order to return to the selection panel.

When you terminate CICS-WINDOWS with a WNDO,OFF command, CICS-WINDOWS checks all virtual terminals to insure that ICCF is not active in any one of them, and will not allow you to WNDO,OFF if it is. If you could terminate CICS-WINDOWS with an active ICCF session, you would not be able to log back on to ICCF with that Operator ID until CICS is recycled.

A somewhat confusing situation can exist if the following sequence of events occurs:

1). ICCF is entered from a selection panel in any virtual terminal, let's say terminal number 2.

2). PF3 is then pressed (still in the same virtual terminal), causing a return to the selection panel. At this point, and automatic log-off of ICCF has not yet occurred, and will not occur until some non-ICCF function is invoked from the panel, such as exiting the panel with PF6.

3). The toggle key is pressed to move to another virtual terminal (terminal 3).

4). A WNDO,OFF command is issued in terminal 3.

5). Message WNDO0010 is displayed indicating that ICCF is still active in terminal 2. This does not appear to be the case, since the selection panel is displayed in terminal 2, but in fact it is true because the automatic log-off for ICCF has not occurred.

    In order to complete the WNDO,OFF, you must return to virtual terminal 2, exit the panel using PF6 (or any non-ICCF selection). This will cause ICCF to be logged-off. You may then issue the WNDO,OFF command and it will complete.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

OVERHEAD CONSIDERATIONS WITH MULTIPLE INTERACTIVE INTERFACE SESSIONS

We at Unicom Systems don't want to discourage you from using multiple Interactive Interface sessions with CICS-WINDOWS because we think it is a nice feature. However, you should know that there is a price to pay in system overhead.

The Interactive Interface (without CICS-WINDOWS involved) operates pseudo-conversationally by allocating a buffer of static storage from the Shared storage subpool to each terminal using a selection panel. This buffer varies in size depending on the number of selections on the panel, but the size of the buffer for the system supplied panel is just over 2K. This storage does not get released until the operator signs off. Nor does it get written to Temporary Storage or to any file. It stays in virtual memory.

When you use multiple selection panels with CICS-WINDOWS, a buffer of the same size is allocated from the Shared subpool for each virtual terminal. Thus, a physical terminal with four virtual terminals and a selection panel in each session could tie up as much as 8k of the Dynamic Storage Area of CICS.

The only way to free these areas is to sign off from CICS. Thus, it might be preferable to use the VSE INTERACTIVE INTERFACE=SIGNON option and assign multiple sign-ons to each operator, rather than use the VSE INTERACTIVE INTERFACE=YES option. With the VSE INTERACTIVE INTERFACE=YES option, the buffer from the first virtual terminal is copied into all virtual terminals at WNDO,ON time. With the VSE INTERACTIVE INTERFACE=SIGNON option, if the operator has not signed on in a virtual terminal, no storage will be allocated.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 14. SPECIAL-PURPOSE COMMANDS

A series of special commands are available with CICS-WINDOWS which are useful in controlling operation, installing new versions, problem solving, etc. Among these are commands to:

- Back out the 'hooks' in CICS to allow a new copy of the programs to be loaded. (BACKOUT)
- Stop anyone from activating CICS-WINDOWS at a terminal. (STOP)
- Restart CICS-WINDOWS operation (allow activation at a terminal). (START)
- Switch from using Temporary Storage to using MAIN storage above the line for saving data. (MAIN)
- Switch from using MAIN to Temporary Storage for saving data. (TEMP)
- Force deactivation of CICS-WINDOWS at a terminal or all terminals. (PURGE)
- Activate CICS-WINDOWS at a terminal, bypassing the Auto-start table. (INIT)
- Produce a series of transaction dumps to aid Unicom Systems Technical Support in solving a problem. (DEBUG)
- Turn the Data Compression feature on or off. (CPRON,CPROFF)

[Note]: All of the Special Purpose commands can be secured so that only authorized personnel can issue them. See the discussion on command security in section 11 – *CUSTOMIZATION*, under THE *USER OPTIONS TABLE*.

COMMAND SYNTAX AND USAGE

BACKOUT

Command syntax: WNDO,**BACKOUT**

The BACKOUT command can be used when a new version of CICS-WINDOWS needs to be installed, a PTF applied or for any reason a new copy of either the WINDOWS, WNDOMAIN, or WNDOTBL programs need to be loaded into CICS.

[Note]: If both the WNDO,STOP command the WNDO,BACKOUT command are issued, a new copy must be performed before CICS-WINDOWS can be reactivated.

The BACKOUT command performs the following functions when invoked:

1) All intercept points ('hooks') in CICS are removed and the normal CICS addresses are replaced.
2) The Terminal Control User Exits are disabled.
3) The 'in-use' condition for the two modules WINDOWS and WNDOTBL is set to zero, which allows a CEMT NEWCOPY command on these modules to be performed.

The end result of the BACKOUT command is to put CICS-WINDOWS back to the same status as if CICS had just been initialized and no one had issued any WNDO command.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The BACKOUT command can only be issued if all users have deactivated CICS-WINDOWS at their terminals by issuing a WNDO,OFF command. In order to prevent anyone from performing a WNDO,ON while BACKOUT and NEWCOPY are being performed, you may want to issue the WNDO,STOP command (described below) before issuing the BACKOUT.

CPROFF

Command syntax: WNDO,**CPROFF**

The CPROFF command is to be used to deactivate the Data Compression feature of CICS-Windows. This command could be used any time it is desired to turn data compression off for all terminals, regardless of the coding in the User Option Table.

CPRON

Command Syntax: WNDO,**CPRON[,YES]**

The CPRON command is to be used to activate the Data Compression feature of CICS-Windows. This command could be used if a CPROFF command has previously been issued, or if DATA STREAM COMPRESSION is not selected in the User Option Table.

DEBUG

Command syntax: WNDO,**DEBUG,xxxx,ON|OFF**

The DEBUG command is to be used on request by UNICOM Systems, Inc. Technical Support when some sort of technical problem has arisen with CICS-WINDOWS. The DEBUG command causes a series of transaction core dumps to be output at strategic points in CICS-WINDOWS operation. To use the DEBUG command, enter the following:

WNDO,DEBUG,xxxx,ON

where xxxx is the terminal ID of the terminal to be used in recreating the problem situation. Note that you cannot issue the DEBUG command for the terminal with the problem. It must be issued from another terminal. After executing the transaction which causes the problem on terminal xxxx, enter the following:

WNDO,DEBUG,xxxx,OFF

This terminates DEBUG mode for terminal xxxx. The transaction dumps will be in the Dump Dataset. Run the Dump Utility and send the dumps to UNICOM Systems, Inc. Technical Support.

INIT

Command syntax: WNDO,**INIT**

The INIT command can be used instead of a WNDO,ON command when it is desired to activate CICS-WINDOWS without using the values coded in the User Profile for this terminal.

This command will cause the operator to be prompted for each CICS-WINDOWS option.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

When the **WNDO,INIT** command is issued, you will be presented with the following prompts, one line at a time. Your answers to these questions will affect how the application works on your terminal.

If no User Profiles are defined, both the WNDO,ON and WNDO,INIT commands have the same effect; however if a User Profile (see section 11 - *CUSTOMIZATION*) has been defined for this terminal (either explicitly or by default), the **WNDO,ON** command will bypass some or all of the following prompts.

After CICS-WINDOWS has been initiated by entering "WNDO,INIT", you will be prompted with:

==> *SELECT* NUMBER OF TERMINALS

Key in a number from 2 to 9 for the number of concurrent applications or sessions (virtual terminals) that you desire to have available on this physical terminal and press ENTER. Your choice should be based on the number of terminals you feel will be needed throughout the day. Each virtual terminal is capable of operating any CICS application. One rule of thumb is to select one more terminal than the average number of different applications that you use in a day.

You will next be prompted with:

==> *PRESS DESIRED TOGGLE KEY*

This key will be used to move you sequentially from one virtual terminal to another in a "forward" direction. Press any program function (PF) or program attention (PA) key. This is now the key to press to move from the current virtual terminal to the next. (You should select a key that is not used as a PF or PA key in your applications. We suggest using a PF key for this function.)

The next prompt will be:

==> *SELECT WINDOWING (Y/N)*

This question is asked because you do have a choice with CICS-WINDOWS. If windowing is desired, press ENTER (the default). Otherwise enter 'N'.

If you said 'N', the next prompt is bypassed, otherwise it is:

==> *PRESS DESIRED WINDOW KEY*

This key will be used to switch the terminal between full-screen and "window" mode. Press any program function (PF) or program attention (PA) key. The key you select will become the key used to go to and from the window mode. This key works as an on/off toggle switch; the first time it is pressed puts the terminal into window mode. The next time pressed it puts the terminal into full-screen mode.

The next prompt you will receive is:

==> *SELECT NUMBER OF WINDOWS*

This question gives you a choice of how many windows will appear on your terminal and their initial configuration. You will note that there are two one-position fields following the prompt. The first position is the number of windows that will be available on your screen when you press the "window" key. Type a '2', '3' or '4' in the first field.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The second position pertains only to a two-window configuration. It gives you the option to select vertical or horizontal division of the screen as the initial configuration. If you typed '2' in the first field, place a 'V' (vertical) or 'H' (horizontal) in the second field ('H' is the default).

The initial size and position (configuration) of the windows depends on the number of windows selected. For a description of the various window configurations and examples of each, see section 04 - *USING THE WINDOWING FEATURE*.

The final prompt that you will receive is:

```
==> ENTER PSEUDO TERMINAL ID'S IF DESIRED
      (1)     (2)     (3)     (4)     (5)
     xxxx    xxxx    xxxx    xxxx    xxxx
```

The number of IDs shown on the line will correspond to the number of virtual terminals that you selected. This is an optional entry, read the following to determine if you need to make the entry.

Certain CICS applications are terminal ID dependent, in the sense that they use your physical terminal ID as part of the processing logic. When more than one of these applications is invoked on different virtual terminals on the same physical terminal, there can be contention, which could cause unexpected results.

The Pseudo Terminal option is provided to prevent this contention by allowing the concurrent use of these types of applications. You may want to consult with your MIS system's programmer to decide if you need to use Pseudo IDs and, if so, what value should be assigned to each one. (Note for MIS personnel: see the discussion entitled *PSEUDO TERMINAL ID CONSIDERATIONS* in section 12 - *SPECIAL CONSIDERATIONS*).

To invoke the Pseudo Terminal feature, do the following:

1) The (1), (2), (3), etc. under the message refers to each virtual terminal that you have selected. The xxxx is pre-filled with the real ID of this physical terminal. You cannot change the ID of terminal number one (1). Enter a unique ID in the ID positions under the remaining virtual terminal positions. The ID must be unique. CICS-WINDOWS will tell you if it is not. If you wish for CICS-WINDOWS to generate the pseudo terminal IDs, enter "*GEN" in the field for terminal 2.

2) After you have keyed the ID's, press ENTER.

If Pseudo Terminals are not desired or needed, simply press ENTER without modifying the terminal IDs. This will cause all ID's to be the same as the physical terminal ID.

At this point, CICS-WINDOWS initialization is complete and the User Configuration Display will be displayed next.

MAIN

Command syntax: WNDO,MAIN

The MAIN command will cause CICS-Windows to switch from using Temporary Storage to using MAIN storage above the 16M line for saving terminal data.

PURGE

Command syntax: WNDO,**PURGE,xxxx**|**ALL[,FORCE]**

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The PURGE command can be used to force ICIS-WINDOWS to deactivate from a terminal. The terminal ID to be purged is specified following the PURGE command. Or, if ALL is specified as the terminal ID, all terminals using CICS-WINDOWS will be purged.

PURGE,ALL should be used only in emergency situations when it is necessary to quickly remove all users from CICS-WINDOWS for any reason.

PURGE could be used when it is not possible to issue a WNDO,OFF at the terminal, a terminal error has placed the terminal out of service, for instance. Or perhaps the terminal is unattended and a BACKOUT command needs to be performed.

For DOS users, if ICCF is active in any session except the current session for a terminal, the purge will not take effect unless the command is followed by ‘,FORCE’. The PURGE,ALL function will stop with message WNDO0048 when a terminal with ICCF active is found. For VSE users, using the FORCE operand can cause an outstanding ICCF session which cannot be logged off. Nor can you log back on to ICCF using the same Operator ID until CICS is recycled.

The PURGE commands (single or ALL) perform the following functions when issued:

1). The terminal is removed from the chain of active WINDOWS terminals and the storage occupied by the chain entry is released.

2). The terminal ID in the TCTTE is reset to the real ID (if Pseudo IDs were in use).

3). Any conversational tasks which were toggled out of for that terminal are ABENDed with an AKCS abend code, and all storage occupied by the task is released.

Usually, the purge function will have no effect on the current transaction in progress on that terminal. The operator may not be aware that a purge was done until he or she attempts to use one of the toggle or window keys, at which time it will be inoperative.

START

Command Syntax: WNDO,**START**

The START command cancels a previous STOP command and allows users to perform a WNDO,ON command to activate CICS-WINDOWS at their terminal.

[Note]: If both the WNDO,STOP command and the WNDO,BACKOUT command are issued, the WNDO,START command will not restart CICS-WINDOWS. A new copy must be performed before CICS-WINDOWS can be reactivated, in which case, the WNDO,START command will not be necessary.

STOP

Command syntax: WNDO,**STOP**

The STOP command will prevent anyone from performing a WNDO,ON command until a subsequent WNDO,START command is issued. The STOP command should be used anytime a BACKOUT, MAIN, or TEMP command needs to be issued.

[Note]: If both the WNDO,STOP command and the WNDO,BACKOUT command is issued, a new copy must be performed before CICS-WINDOWS can be reactivated.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TEMP

Command syntax: WNDO,**TEMP**

The TEMP command will cause CICS-WINDOWS to switch from using MAIN storage to using Temporary Storage for saving terminal data.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 15. USER EXITS AND PROGRAM INTERFACES

CICS-WINDOWS can be customized still further by writing one of the available user exit programs or using one of the interface points where CICS-WINDOWS can be called from a user program.

This section deals with all of the available user exit and program interface points. Each routine is described, along with the reason that you might want to use it, then a coded example is illustrated. Many of these example routines and programs are distributed on the installation tape. They are present in the first file, which contains all of the source code that is distributed with the product.

Most user exits and interface routines in this section are coded in Assembler language. The user should have a good working knowledge of Assembler and CICS coding conventions for both macro level and command level.

[Note]: For versions 3.2 and 3.3 of CICS, it is still possible to do all the macro-level functions described herein, and reference both the CSA and TCA. For more details on this, see *PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3*, later in this section.

WRITING A USER EXIT FOR CICS-WINDOWS

There are two types of user exit programs that can be written to alter the way that CICS-WINDOWS performs in certain situations. These are:

1) Transaction attach exit.
2) Pseudo Terminal ID generation exit.
3) Terminal deactivation exit.

The exit program must be written in Assembler language. It must be assembled and the object module cataloged in a library or PDS where it can be included with the WNDOTBL module during Link-Edit. In addition, the presence of the exit program must be indicated in the WNDOTBL macro by means of a keyword specifying the exit program name.

You can perform any CICS functions in the user exit program, so long as they are coded in Macro-Level CICS. The exit program can not perform any Command-level functions directly, however it is possible to do a DFHPC TYPE=LINK to a Command-level program that is specified in the PPT.

You can address the terminal (TCTTE) by loading field TCAFCAAA, and obtain any information desired from it. You may need the terminal ID (TCTTETI), the Operator ID (TCTTEOI), the security keys (TCTTESK and TCTESKE) or possibly the terminal I/O area (TCTTEDA). CICS 1.7 users can obtain the Operator ID and other sign-on information by accessing the Sign-on table block addressed by field TCTTESNT. The DSECT for this area can be obtained by coding DFHSNT TYPE=DSECT.

If you need a dynamic work area for your exit program you can either perform a GETMAIN or define a TWASIZE parameter in the PCT for the WNDO transaction and use the Transaction Work Area. CICS-WINDOWS will not use a TWA itself and your exit program will be running under the same TCA as the WINDOWS program.

Each of the user exits, along with coding details, is described below.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TRANSACTION ATTACH EXIT

The Attach exit will get control when CICS-WINDOWS is ready to initiate a transaction in window mode. The exit program can decide whether to allow the requested transaction to run or alter it to some other transaction.

If you are currently using a Terminal Control Global User Exit to CICS which gets control at transaction attach time, you must code the CICS-WINDOWS Attach exit in order to perform the same function for transactions attached in window mode. This is because CICS-WINDOWS uses a Task Control attach macro to initiate a task. The Global Terminal Control exit will not be invoked again.

You activate this exit by coding ATCHXIT=xxxxxxx in the WNDOTBL macro (see section 11 - *CUSTOMIZATION*). The xxxxxxxx is a 1 to 8 character name of your program. It must be the same as the name on the CSECT or START statement of your program.

The exit is constructed to use roughly the same linkage conventions as the Global attach exit to CICS. Specifically, the convention is as follows:

R15	-	Contains the entrypoint to the exit program.
R14	-	Contains the return address.
R13	-	Contains a 72-byte savearea address.
R1	-	Contains the address of a parameter list.

The parameter list pointed-to by R1 can be described using the DFHUEPAR DSECT, as in a Global CICS exit.

The parameter list contains the following fields:

Position 0	-	Four fullwords containing hex zeros.
Position 16	-	Address of the TCA.
Position 20	-	Address of the CSA.
Position 24	-	Address of the caller's savearea.
Position 28	-	Fullword of zeros.

Field TCAKCTI in the user TCA contains the transaction code that is ready to be attached.

There is no return code to be set. The only thing that the exit program can do to affect operation is to alter the transaction code in TCAKCTI. CICS-WINDOWS will subsequently attach whatever transaction is set.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ATTACH EXIT EXAMPLE

Following is an example of a transaction attach user exit. This example includes DOS JCL to assemble the exit and catalog it to a relocatable library.

```
// JOB ASSEMBLE ATTACH EXIT
// OPTION DECK,NOSYM
// DLBL  IJSYSPH,'SYSPUNCH.FILE',0
// EXTENT SYSPCH,VOL2EC,1,0,100,20
ASSGN SYSPCH,2EC
// EXEC ASSEMBLY,SIZE=200K
   TITLE   'WINDOWS ATTACH EXIT'
   PUNCH  '  CATALR  ATCHWIN'
   COPY    DFHTCADS       TCA DSECT
TCTTEAREQU     10
   COPY    DFHTCTTE       TCTTE DSECT
   DFHUEXIT  TYPE=EP    PARMLIST DSECT
   EJECT
ATCHWIN          CSECT
   USING  *,15
   USING  DFHUEPAR,1    ADDRESS PARMLIST
   STM     14,12,12(13)       SAVE CALLER REGS
   L          12,UEPTCA       LOAD TCA ADDRESS
   L          TCTTEAR,TCAFCAAA   LOAD TCTTE ADDRESS
   CLC     TCTTEOI,=C'LGA'         IS OPERATOR 'LGA' ?
   BNE     ATCHRET             NO, EXIT
   CLC     TCAKCTI,=C'CSMT'       IS TRANCODE CSMT ?
   BNE     ATCHRET             NO, EXIT
   MVC     TCAKCTI,=C'CEMT'       MAKE IT CEMT
   SPACE
ATCHRET          EQU     *
   LM       14,12,12(13)       RELOAD CALLER REGS
   BR       14         RETURN TO CICS-WINDOWS
   LTORG
   END     ATCHWIN
/*
CLOSE  SYSPCH,PUNCH
// DLBL  IJSYSIN,'SYSPUNCH.FILE',0
// EXTENT SYSIPT
ASSGN  SYSIPT,2EC
//  EXEC  LIBR,PARM='A S=xxx.xxx'
/*
CLOSE  SYSIPT,READER
/&
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PSEUDO TERMINAL ID GENERATION EXIT

This exit will get control when CICS-WINDOWS is ready to generate the Pseudo Terminal IDs during initialization processing as a result of a WNDO,ON or INIT command.

You activate this exit by coding PSGNXIT=xxxxxxx in the TYPE=INITIAL statement of the WNDOTBL macro (see section 11 - *CUSTOMIZATION*). The xxxxxxxx is a 1 to 8 character name of your program. It must be the same as the name on the CSECT or START statement of your program.

Upon entry to the exit program, the following register conventions are used:

R15	- Contains the entrypoint to the exit program.
R14	- Contains the return address.
R13	- Contains a 72-byte savearea address.
R1	- Contains the address of a parameter list.

The parameter list contains the following fields:

Position 0	- Address of a 4-byte field containing the real terminal ID.
Position 4	- Address of a 32-byte area where the exit program is to build the terminal IDs.
Position 8	- Address of a 1-byte hex field containing the number of IDs to build.
Position 12	- Address of the CSA.
Position 16	- Address of the TCA.

The exit program must build the terminal IDs corresponding to virtual terminals 2 thru 9 in the 32-byte area pointed-to by the second word of the parmlist. The terminal IDs must be in increments of 4 bytes each. When complete, bytes 1-4 will be the ID for session 2, bytes 5-8 will contain the ID for session 3, bytes 9-12 will contain the ID for session 4, etc.

You can build all eight pseudo terminal IDs if desired, even though the operator may be using less than nine virtual terminals. Or, if you prefer, use the 1-byte hex value pointed-to by the third word of the parmlist to construct only the number of IDs needed. Keep in mind that a WNDO,INIT command bypasses the Auto-Init table entries, so it is possible for an operator to request more sessions than their User Profile would allow.

Your exit program can decide, if desired, that this operator or this terminal should not use Pseudo IDs at all. In that case, simply move the real ID into all ID positions.

Upon return from the exit program, CICS-WINDOWS will validate that the IDs that your program built are unique among all other real and Pseudo IDs and send the normal WNDO1402 message to the terminal if they are not.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PSEUDO TERMINAL ID GENERATION EXIT EXAMPLE

Following is an example of a Pseudo ID generation user exit. This example includes MVS JCL to assemble the exit and place the object module in an object PDS.

```
//PSGNXIT         JOB     1.'ACCOUNT-DATA',MSGCLASS=x
//ASM     EXEC    PGM=IEV90,PARM='DECK'
//SYSLIB  DD      DSN=CICS.MACLIB,DISP=SHR
// DD     DSN=SYS1.MACLIB,DISP=SHR
//SYSPRINT        DD      SYSOUT=*
//SYSLIN  DD      DUMMY
//SYSUT1 DD       UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH        DD      DSN=CICS.OBJLIB(PSIDGEN),DISP=SHR
//SYSIN   DD      *
  TITLE   'WINDOWS PSEUDO ID GENERATION EXIT'
PSIDGEN CSECT
  USING  *,15
  STM    14,12,12(13)      SAVE CALLER REGS
  LM     4,6,0(1)  LOAD 1ST  3  ADDRESSES
  USING  TERMID,4          ADDRESS REAL TERM ID
  USING  GENAREA,5         ADDRESS ID BUILD AREA
  USING  COUNT,6           ADDRESS ID COUNT FIELD
  MVC    GENAREA(4),TERMID   MOVE REAL ID
  MVC    GENAREA+4(28),GENAREA   PROPAGATE ALL IDS
  XR     0,0
  IC     0,COUNT           LOAD COUNT
  LA     1,GENAREA         R1 = SESSION 2 ID
  LA     2,=C'ABCDEFGH'          R2 = TABLE OF MODIFIERS
GENLOOP         EQU     *
  MVC    0(1,1),0(2)       MODIFY FIRST BYTE OF ID
  LA     1,4(1)    BUMP TO NEXT ID
  LA     2,1(2)    BUMP TO NEXT MODIFIER
  BCT    0,GENLOOP         DO ALL REQUIRED IDS
  SPACE
  LM     14,12,12(13)      RELOAD CALLER REGS
  BR     14        RETURN TO CICS-WINDOWS
  SPACE
  LTORG
  SPACE 3
TERMID  DSECT
  DS     CL4       REAL TERMINAL ID
GENAREA         DSECT
  DS     CL32      AREA FOR 8 PSEUDO IDS
COUNT   DSECT
  DS     XL1       NUMBER IF IDS REQUIRED
  END    PSIDGEN
/*
/&
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INSTALLING THE USER EXIT PROGRAM

To install your exit program, assemble it with the DECK option and place the object in an object library, as in the examples above. For DOS this would be a CATALR into a relocatable library (name.OBJ for VSE users) and for MVS, the object module should go in an OBJLIB or MODLIB PDS.

Now code the program name in the appropriate exit operand in the User Option Table (ATCHXIT or PSGNXIT or both), then assemble and link-edit WNDOTBL. If the library module name of your exit program is the same as the CSECT or START name, the Linkage Editor will auto-link the exit program with WNDOTBL. Otherwise you must have an INCLUDE statement in the Link-Edit step to retrieve the module.

Check the output from the Linkage-Editor to make sure that you have no unresolved address constants.

Now issue a WNDO,BACKOUT command, then perform a CEMT NEWCOPY for WNDOTBL. The exit program is now installed and will take effect when CICS-WINDOWS is activated.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE MENU GENERATION FEATURE USER EXIT

The menu generation user exit is used to edit the available selections on a menu. It is called once for each of the selections available on the menu immediately prior to displaying the menu.

The exit program may perform any of the following functions:

- allow the menu selection. (do nothing)
- blank the menu selection from the menu display
- delete the menu selection from screen, and shift all other menu selections up one line
- display the menu selection on the screen, but disable the selection
- alter the selection parameters (i.e. change the transaction code of a transaction to be initiated)
- alter the text that is displayed for a selection

A sample exit program and parameter copybook is distributed on the installation tape. The program name is MENUEXIT and the copybook is CWxx$M32 (where xx is the release of CICS-WINDOWS.

The exit program is passed a parameter list in the commarea. The parmlist contains 3 addresses as described below:

Menu Selection Definition

This is the single selection definition that applies to this call of the exit program. The format of this definition is shown below, and also in the CWxx$M32 source copybook (where xx is the release of CICS-WINDOWS.)

```
MNDDSECT        DSECT
                DS      XL1     RESERVED
MNDMNUID        DS      CL8     MENUID
                DS      XL1     RESERVED
MNDEXIT         DS      CL8     USER EXIT ID
                DS      XL1     RESERVED
MNDQUIT1        DS      CL1     ALLOW QUIT AT LEVEL ONE MENU CONTAINS:
MNDQYES         EQU     C'/'            TO ALLOW QUIT
MNDQNO          EQU     C''             TO NOT ALLOW QUIT
                DS      XL1     RESERVED
MNDREFS         DS      CL1     REFRESCH MENU AT TRANSACTION END CONTAINS:
MNDRYES         EQU     C'.'            TO REFRESH MENU
MNDRNO          EQU     C''             TO NOT REFRESH MENU
                DS      XL4     RESERVED
MNDACTN         DS      CL1     ACTION CODE  CONTAINS:
MNDAOK          EQU     C'O'            ALLOW MENU SELECTION
MNDABLNK        EQU     C'B'            BLANK MENU SELECTION FROM MENU
MNDADEL         EQU     C'D'            DELETE SELECTION FROM MENU
MNDAIGN         EQU     C'I'            DISPLAY SELECTION BUT DON'T
                                        ALLOW USER TO ACCESS IT
                DS      XL1     RESERVED
MNDENTRY        DS      CL2     SELECTION ENTRY NUMBER
                DS      XL1     RESERVED
MNDTYPE         DS      CL4     SELECTION TYPE CONTAINS:
MNDTTRAN        EQU     C'T'            C'TRAN' FOR TRANSACTION
MNDTPROG        EQU     C'P'            C'PROG' FOR PROGRAM
MNDTQUIT        EQU     C'Q'            C'QUIT' FOR QUIT
MNDTESC         EQU     C'E'            C'ESC' FOR ESCAPE
MNDTMENU        EQU     C'M'            C'MENU' FOR MENU
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
MNDTSUBM    EQU   C'S'        C'SUBM' FOR SUBMENU
            DS    CL1   RESERVED
MNDID       DS    CL8   PROGRAM, TRANSACTION OR MENU ID
            DS    XL1   RESERVED
MNDPFKEY    DS    CL4   PFKEY MNEMONIC
            DS    XL1   RESERVED
MNDROW      DS    CL2   CURSOR SELECTION ROW
            DS    XL1   RESERVED
            DS    XL1   RESERVED
MNDCOL      DS    CL2   CURSOR SELECTION COLUMN
            DS    CL1   RESERVED
MNDINPUT    DS    CL8   SELECTION INPUT
            DS    CL1   RESERVED
MNDSTYPE    DS    CL6   START TYPE (FOR TRANSACTION OF PROGRAM) CONTAINS:

MNDSATCH    EQU   C'A'        C'ATTACH' FOR ATTACH TRANSACTION

MNDSSTRT    EQU   C'S'        C'START' FOR START TRANSACTION

MNDSLINK    EQU   C'L'        C'LINK' FOR LINK TO PROGRAM

MNDSXCTL    EQU   C'X'        C'XCTL' FOR XCTL TO PROGRAM

            DS    XL1   RESERVED

MNDSWCMD    DS    CL8   WINDOW COMMAND

            DS    XL1   RESERVED

MNDSDATA    DS    CL76  DATA TO BE PASSED TO PROGRAM/TRANSACTION

            DS    XL1   RESERVED

MNDWORK     DS    XL80  WORK AREA
```

Any of these fields (except the reserved fields and MENU ID) may be modified by the exit program. These fields are explained in the following paragraphs.

The MNDACTN field may be used by the exit program to alter the way the menu is displayed or executed. The exit program may pass any of the following values to the menu generator:

- O Allow the menu selection.
- B Blank the menu selection, so that it will not display on the menu and disable the selection so that it may not be selected.
- D Delete the selection from the screen and disable the selection so that it may not be selected.. This will move all subsequent lines up to fill the deleted selection.
- I Ignore the selection. This will display the selection on the menu but will not allow this option to be selected.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The MNDWORK field is not used or altered by the menu generator. It is provided as a sort of common area between calls of the exit program.

The remainder of the fields correspond directly to the Menu Selection Definition Display. For more information on these fields, refer to section 08 – *MENU GENERATION*.

Menu Selection Text

This is the address of the actual menu option text as it appears on the menu display. You may change the text that is to display. However, this address points to an area within the entire menu display. This should be modified with caution since the modification could corrupt the remainder of the menu display. For more information, see the description of the Menu Display, below.

Menu Display

This is the menu screen display in a ‘flat record’ type format. For a model 2 terminal, this will be an area of storage that in 1920 bytes long. All display attributes, whether extended or regular, are represented by a single-byte internal attribute code. Some of the internal attribute codes are displayable characters, but none are characters that can be keyed from a standard U.S. English keyboard.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

To install the menu generation feature user exit, assemble and link the program to a CICS load library, and create a PPT entry (or for RDO a program entry). Then enter the name of the exit program on the appropriate Menu Definition Display, in the USER EXIT field.

PROGRAM INTERFACE ROUTINES

You can call CICS-WINDOWS from a user program to perform various functions. Some of these are fairly specialized functions, but most are routines that have been developed in response to the needs described by other users of the product. Depending on your environment, you may find that you need to develop one or more of these interface calls in order for CICS-WINDOWS to operate properly in all circumstances.

Some of the uses of the program interface routines are summarized as follows:

1) Activate or Deactivate CICS-WINDOWS at a terminal.

2) Perform session and windowing commands under program control.

3) Deactivate CICS-WINDOWS at a terminal when a node error occurs.

4) Deactivate CICS-WINDOWS at a terminal prior to an automatic log-off, or for any other reason.

5) Retrieve information about the current session in progress.

6) Obtain the real terminal ID from a pseudo ID.

7) Test to see if CICS-WINDOWS is active on a terminal.

8) Accessing or altering the active user table entry.

9) Change the transaction code to be initiated in a recurring transaction start table.

There are two types of interfaces to CICS-WINDOWS. The first technique is known as the APPLICATION PROGRAM INTERFACE (API) and consists of a direct program LINK, with a command, either in a terminal I/O area or in the TWA. With this method you can perform virtually any command that can be entered at a terminal by an operator.

The second type of interface is a series of special-purpose functions whereby a user program can communicate with CICS-WINDOWS by means of a direct call. These routines are called the BUILT-IN FUNCTIONS, and are described following the API technique, later in this section.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE APPLICATION PROGRAM INTERFACE

An application interface to CICS-WINDOWS is provided whereby the WINDOWS program may be called from a command level or macro level CICS Application program and the various commands and functions of CICS-WINDOWS may be invoked under program control.

This feature enables programmers to take advantage of the features of CICS-WINDOWS when developing or enhancing their Application Systems. Examples of effective use of the Application Interface would be:

1) Automatically turning CICS-WINDOWS on for a terminal, from a sign-on program or a menu screen, for instance.

2) Automatically turning CICS-WINDOWS off for a terminal.
3) Initializing each virtual terminal with a particular Application Program.
4) Windowing Applications together under program control.
5) Routing a display from one window to another.

FUNCTIONS AVAILABLE

CICS-WINDOWS, when linked-to by an application program, will perform the functions based on a command stream defined by the calling program. The command name corresponding to each function is as follows:

- Activate (WNDO,ON) for this terminal
- Deactivate (WNDO,OFF pr WNDO,PURGE) for this terminal
- Toggle from the current session to the next session
- Toggle from the current session to the previous session
- Enter window mode
- Exit window mode
- Set scrolling position for a window
- Expand a window to full screen mode
- Designate a window to receive the next display
- Clear a window
- Designate the active window for transaction input
- Remove CICS-WINDOWS from the system for a NEWCOPY
- Purge a single (or all terminals)
- Start CICS-WINDOWS after a previous stop command
- Stop all users from activating CICS-WINDOWS.

In effect, most of the terminal input commands to CICS-WINDOWS will function under program control, with the additional capability that they may be "streamed". That is, a series of windowing commands may be interspersed with links to Application Programs.

For instance, the streaming sequence to activate CICS-WINDOWS, start a user application in three sessions, then enter window mode would be as follows:

1) Link to WINDOWS with WNDO,ON command. If the User Profile for that terminal or operator has all necessary options defined, the user will receive the User Configuration Display, otherwise the operator must respond to the WNDO,ON prompts.
2) Link to an application Program for virtual terminal number 1. The operator will see the display in full-screen mode from this application.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

3) Link to WINDOWS with a "toggle" command to move to virtual terminal number 2.
4) Link to another (or the same) application Program. The display from that application will appear.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

5) Link to WINDOWS with a "toggle" command to move to virtual terminal number 3.
6) Link to the application Program.
7) Link to WINDOWS with a "WIN,IN" command. The operator will be presented with the window mode display containing a partial display from each application program.

LINKING TO CICS-WINDOWS

To invoke the application interface, the calling program must issue a CICS Link command to the program "WINDOWS".

In command level, the format of the Link statement is:
EXEC CICS LINK PROGRAM(WINDOWS) END-EXEC.

In macro level, the format is:
DFHPC TYPE=LINK,PROGRAM=WINDOWS

PASSING COMMANDS

Commands are passed to the WINDOWS program starting in position 1 of the Transaction Work Area (TWA). This area may be addressed in a Command Level program with the statement:

EXEC CICS ADDRESS TWA(ptr-ref) END-EXEC.

Positions 1-5 of the TWA must contain the characters 'WNDO,'. Beginning in position 6, one of the following commands may be used:

1) AWINx - Designate the session number to be the active window for the next terminal input.
2) BACKOUT - Remove CICS-WINDOWS from the system in order to perform a NEWCOPY.
3) CWINx - Clear window number x.
4) EXPNx - Expand a designated window to full-screen mode.
5) ON - Activate CICS-WINDOWS on this terminal.
6) OFF - Deactivate CICS-WINDOWS on the terminal.
7) PAN,Wx,Ryy,Czz - Set scrolling position, where:
   x = Session number.
   yy = Row number to start display.
   zz = Column number to start display.
8) PURGE,ALL - Purge CICS-WINDOWS from all terminals.
9) PURGE,xxxx - Purge CICS-WINDOWS from terminal xxxx.
10) RWINx - Designate a session number (x) to receive the next output display, whether it comes from the calling program or from another application program.
11) START - Start CICS-WINDOWS. (Used after a previous STOP command)
12) STOP - Stop all users from activating CICS-WINDOWS on their terminals.
13) TB - Perform a Toggle Backward to the previous session.
14) TF - Perform a Toggle Forward to the next session.
15) Tx - Perform a toggle directly to session x.
16) WIN,IN - Enter window (split-screen) mode.
17) WIN,OUT - Exit window mode, to full-screen mode. Control will pass to the virtual terminal that was in control when window mode was last entered.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

[Note]: In 1, 3, 7 and 10 above (marked with †), x refers to the virtual terminal number, not the window number. The action will take effect in whatever window contains the session data for that virtual terminal.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

APPLICATION INTERFACE EXAMPLE

The following example illustrates a Command Level Cobol CICS program that exercises the various functions of the application Interface. This program receives a formatted command from the operator, then links to the WINDOWS module to perform it.

```
*****************************************************************
THIS PROGRAM CALLS WINDOWS TO TEST VARIOUS FUNCTIONS.
*****************************************************************

IDENTIFICATION DIVISION.
PROGRAM-ID.          WNDOTST.
AUTHOR.              CAJ.
INSTALLATION.        UNICOM SYSTEMS.
DATE-WRITTEN.        11/11/85.
DATE-COMPILED.
SECURITY.            FREE-ACCESS.

*****************************************************************
   ENVIRONMENT DIVISION.
*****************************************************************

DATA DIVISION.
WORKING-STORAGE SECTION.


01      INQUIRY.
        05      INQ-MSG          PIC  X(13)  VALUE 'ENTER COMMAND'.
        05      ERROR-MSG        PIC  X(12)  VALUE 'ERROR RESPONSE'.
        05      MSG-RESPONSE.
                10  MSG-RESPONSE-FIRST-3    PIC  X(3) VALUE SPACES.
                10  FILLER          PIC  X(11) VALUE SPACES.

01      MSG-LENGTH               PIC  S9(4) COMP VALUE +13.
01      RESP-LENGTH              PIC  S9(4) COMP VLAUE +14.
01      ERROR-MSG-LENGTH         PIC  S9(4) COMP VALUE +12.

LINKAGE SECTION.

01      DFHBLLDS.
        02      FILLER           PIC  S9(8) COMP.
        02      TWA-PTR          PIC  S9(8) COMP.

01      TWA-AREA.
        02      TWA-START.
                03      WND0-COMMANDS            PIC  X(19).
                03      RESPONSE REDEFINES WNDO-COMMANDS.
                        05  FILLER                          PIC  X(5).

                        05  COMMAND-IN                      PIC  X(14).
                03      FILLER REDEFINES WNDO-COMMANDS.
                        05  WNDO-RETURN-CODE  PIC  X.
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
*****************************************************************
PROCEDURE DIVISION.
*****************************************************************
*
******  ESTABLISH ADDRESSABILITY TO TWA
*
                EXEC CICS ADDRESS TWA(TWA-PTR) END-EXEC.

*
*****   REQUEST AND RECEIVE COMMAND FROM OPERATOR
*
        100-START.
                EXEC CICS SEND FROM(INQ-MSG)
                LENGTH(MSG-LENGTH) WAIT END-EXEC.

                EXEC CICS RECEIVE INTO(MSG-RESPONSE)
                LENGTH(RESP-LENGTH) END-EXEC.

*
*****   INTERROGATE AND PERFORM REQUESTED COMMAND
*
        IF MSG-RESPONSE = 'ON' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'OFF' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'T' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'WIN,IN' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'WIN,OUT' GO TO WNDO-LINK.
        IF MSG-RESPONSE-FIRST-3 = 'PAN' GO TO WNDO-LINK.
        IF MSG-RESPONSE-FIRST-3 = 'RWI' GO TO WNDO-LINK.
        IF MSG-RESPONSE-FIRST-3 = 'CWI' GO TO WNDO-LINK.
        IF MSG-RESPONSE-FIRST-3 = 'AWI' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'ALT,OFF' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'ALT,ON' GO TO WNDO-LINK.
        IF MSG-RESPONSE-FIRST-3 = 'EXP' GO TO WNDO-LINK.
        IF MSG-RESPONSE = 'CANCEL' GO TO RETURN-TO-CICS.
                        GO TO 100-START.

*
*****LINK TO THE WINDOWS MODULE
*
WNDO-LINK.
        MOVE 'WNDO' TO WNDO-COMMANDS.
        MOVE MSG-RESPONSE TO COMMAND-IN.
        EXEC CICS LINK PROGRAM('WINDOWS') END-EXEC.
        IF WNDO-RETURN-CODE = HIGH-VALUES
        EXEC CICS SEND FROM(ERROR-MSG)
        LENGTH (ERROR-MSG-LENGTH) END-EXEC
        GO TO RETURN-TO-CICS.

*
*****RETURN TO CICS
*
RETURN-TO-CICS.

        EXEC CICS RETURN END-EXEC.
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INVOKING APPLICATION PROGRAMS

You may invoke an application program at any time during a stream of commands to the WINDOWS program by performing a LINK command for requested applications. You should use LINK rather than START because the calling program is attached to the terminal and CICS will not START another program until the calling program returns to CICS.

The following considerations apply when invoking applications via the LINK command:

1) The calling program must have a TWASIZE in the PCT as large as any program being invoked will need.

2) If the invoked program requires a Terminal I/O area, the calling program must be written in Macro Level or running on CICS version 3 in order to build the TIOA and pass address in TCTTEDA.

3) You may use the COMMEREA of command level to communicate with the invoked program.

4) If the invoked program produces a display to the terminal, the display will appear to user for an instant, until the next command is passed to the WINDOWS program.

5) If the invoked program is a conversational program, the operator will have to respond to it and the application program return to CICS before the next WINDOWS command will be executed.

EXAMPLES OF APPLICATION INVOCATION

The following example is a macro level Assembler program which activates CICS-WINDOWS, invokes an application in the first session, toggles to the next session, invokes another application, enters window mode and designates an active window for terminal input:

```
         TITLE   'TESTDISP - WINDOWS TEST PROGRAM'
*
*****  COPY REQUIRED CICS COPYBOOKS
*
TCTTEAR  EQU     R9
TIOABAR  EQU     R10
         COPY    DFHCSADS
         COPY    DFHTIOA
         COPY    DFHTCTTE
         COPY    DFHTCADS
WNDOCMDS DS      CL14
         EJECT
TESTDISP CSECT
         USING   *,R11
         LR      R11,R14                 SET BASE REG
         L       TCTTEAR,TCAFCAAA        ADDRESS TCTTE
         MVC     WNDOCMDS(14),SPACES     CLEAR COMMAND AREA
         MVC     WNDOCMDS(7),=C'WNDO,ON' SET COMMAND
         BAL     R2,WNDOLINK             TURN ON WINDOWS
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
         SPACE
         BAL       R2,GETTIOA                  GET A TIOA
         MVC       TIOADBA(4),=C'AR10'         SET AR10 TRANCODE
         MVC       TIOADTL,=H'4'                  SET TIOA DATA LENGTH
         DFHPC TYPE=LINK,                         LINK TO ACCOUNTS          X
               PROGRAM=ACCTRECV'                  RECEIVABLE
         SPACE
         MVC       WNDOCMDS(14),SPACES            CLEAR COMMAND AREA
         MVC       WNDOCMDS(11),=C'WNDO,TOGGLE'   SET COMMAND
         BAL       R2,WNDOLINK                    TOGGLE TO SESSION 2
         SPACE
         BAL       R2,GETTIOA                     GET A TIOA
         MVC       TIOADBA(4),=C'AP10'            SET AP10 TRANCODE
         MVC       TIOADTL,=H'4'                  SET TIOA DATA LENGTH
         DFHPC TYPE=LINK,                         LINK TO ACCOUNTS          X
               PROGRAM=ACCTPAY'                   PAYABLE
         SPACE
         MVC       WNDOCMDS(14),SPACES    CLEAR COMMAND AREA
         MVC       WNDOCMDS(11),=C'WNDO,WIN,IN'   SET COMMAND
         BAL       R2,WNDOLINK                    ENTER WINDOW MODE
         SPACE
         MVC       WNDOCMDS(14),SPACES            CLEAR   COMMAND AREA
         MVC       WNDOCMDS(10),=C'WNDO,AWIN1'    SET COMMAND
         BAL       R2,WNDOLINK                    SET WINDOW 1 ACTIVE
         SPACE
         DFHPC TYPE=RETURN                        RETURN TO CICS
         EJECT
WNDOLINK EQU       *
         DFHPC                                    TYPE=LINK,PROGRAM=WINDOWS
         BR        R2                             RETURN TO CALLER
         SPACE 2
GETTIOA  EQU       *
         MVC       TCASCNB,=H'20'                 SET LENGTH NEEDED
         DFHSC TYPE=GETMAIN,                      GET A TIOA                X
               CLASS=TERMINAL,                                             X
               INITIMG=00
         L         TIOABAR,TCASCSA                ADDRESS THE AREA
         ST        TIOABAR,TCTTEDA                SET ADDRESS IN TCTTE
         BR        R2                             RETURN
         SPACE
SPACES   DC        CL14' '
         LTORG
         END
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TESTING SUCCESSFUL EXECUTION

When linking to WINDOWS passing commands in the Transaction Work Area, whether using Macro Level or Command Level, CICS-WINDOWS will pass back a one-byte return code in the first position of the TWA if it is unable to process the command. Either of two return codes may be set:

X'FF' (High Values) - the WINDOWS program will pass back a x'FF' (High- Values) in the first byte of the Transaction Work Area if it is unable to perform the requested command due to a syntactical or logical error in the command as it is found in the TWA.

A syntactical error is simply misspelling or erroneously positioning a command, for instance, WNDO,RWIN 1 instead of WNDO,RWIN1.

Logical errors occur when a command is syntactically correct but the current status of the terminal will not allow the command to be executed. Examples of logical errors are:

1) Specifying window 3 when only two windows are active.
2) Specifying a scrolling position which is beyond the physical boundaries of the screen.
3) Doing a WNDO,WIN,IN command when windowing has been suppressed on the terminal.

X'00' (Low Values) - this code is set only in response to the WNDO,OFF command. It is set if CICS-WINDOWS can not be terminated on the terminal because a task is still active in a session. This could only be a conversational transaction unless the REQUIRE CLEAR SESSIONS BEFORE OFF or REQUIRE TRANSACTION END BEFORE OFF option (see section 11 - CUSTOMIZATION) is in effect. If this condition occurs, CICS-WINDOWS will display message WNDO1409 on the screen prior to returning to the calling program. This situation can be avoided by passing a WNDO,PURGE command for the current terminal rather than a WNDO,OFF.

Note that if a command can not be executed, control simply returns to the calling program and no action is taken. Thus, it causes no problems as far as CICS-WINDOWS is concerned to simply ignore the error return and continue to execute a command stream.

ACTIVATING AND DEACTIVATING CICS-WINDOWS FROM A USER PROGRAM

It is often desirable to start or stop CICS-WINDOWS at a terminal under program control. Many users prefer that CICS-WINDOWS be active on the terminal as soon as the operator signs on. Others use selection menu processing for all applications and prefer to start or stop CICS-WINDOWS at a terminal by having the operator select a menu option. This can be accomplished with no user programming by specifying the AUTOSTART option in the User Profile. If you have special requirements, however, the following technique can be used.

A common practice is to imbed the activation and deactivation of CICS-WINDOWS in the sign-on / sign-off process. A user program is written using transaction code CESN (or any trancode) that first links to the CICS sign-on program, then links to WINDOWS with a WNDO,ON command. The same program could be invoked using transaction code CESF (or any sign-off trancode) that links to WINDOWS with a WNDO,OFF or WNDO,PURGE command, then links or exits to the sign-off program.

Using this method, the sign-on is always performed before CICS-WINDOWS is activated, which means that all sessions will have the same security and no subsequent sign-ons will be required in the virtual terminals. By turning WINDOWS off before signing off, you insure that no sessions are left active when the

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

operator signs-off (this can also be accomplished using the FORCE PURGE AT SIGNON AND SIGNOFF option in the User Option Table).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

By defining the CICS-WINDOWS configuration for the terminal using the customization statements in the User Option Table (see section 11 - CUSTOMIZATION), you can eliminate all operator interaction with CICS-WINDOWS during initialization except for the initial WNDO,ON command. To make the activation and/or deactivation of CICS-WINDOWS totally transparent to the operator, you can invoke the WNDO,ON or WNDO,OFF command from either a Macro Level or Command Level CICS program.

LINKING FROM A MACRO LEVEL PROGRAM

In a Macro Level program, you can use two different techniques to link to the WINDOWS program passing it a command:

1) Obtain a terminal I/O area (TIOA) and construct the command at the start of the TIOA, then perform a Program Control LINK to the WINDOWS program.

2) Build the command starting in the first position of the Transaction Work Area (TWA), then LINK to the WINDOWS program.

With method 1, you can pass any valid WNDO command to CICS-WINDOWS, since the WINDOWS program does not recognize that it is being called from another program in this case. It assumes that the incoming command in the TIOA was entered at the terminal. You can not pass the toggle or window key, however. It must be one of the transaction commands listed in APPENDIX A.

With method 2, the WINDOWS program recognizes the WNDO command at the beginning of the TWA and processes that command as if it had been entered at a terminal.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Macro Level Assembler Examples

Following is an example of method 1, using Macro Level Assembler language:

```
            TITLE   'ACTIVATE CICS-WINDOWS FROM A TIOA'
TCTTEAR     EQU     9
TIOABAR     EQU     10
            COPY    DFHCSADS
            COPY    DFHTIOA
            COPY    DFHTCTTE
            COPY    DFHTCADS
            EJECT
WNDOCALL    CSECT
            USING   *,11
            LR      11,14                   SET BASE REG
            L       TCTTEAR,TCAFCAAA        ADDRESS TCTTE
            MVC     TCASCNB,=H'8'           SET LENGTH NEEDED
            DFHSC TYPE=GETMAIN,             GET A TIOA                  X
                    CLASS=TERMINAL,                                     X
                    INITIMG=00
            L       TIOABAR,TCASCSA         ADDRESS THE AREA
            ST      TIOABAR,TCTTEDA         SET ADDRESS IN TCTTE
            MVC     TIOADBA(7),=C'WNDO,ON'  SET COMMAND
            MVC     TIOADTL,=H'7'           SET TIOA DATA LENGTH
            DFHPC TYPE=LINK,                LINK TO THE                 X
                    PROGRAM=WINDOWS             WINDOWS PROGRAM
            SPACE
            DFHPC TYPE=RETURN               RETURN TO CICS
            END
```

Following is an example of method 2, using Macro Level Assembler language:

```
            TITLE   'DEACTIVATE CICS-WINDOWS'
            COPY    DFHCSADS
            COPY    DFHTCADS
            EJECT
WNDOCALL    CSECT
            USING   *,11
            LR      11,14                   SET BASE REG
            MVC     TWACOBA(8),=C'WNDO,OFF' SET COMMAND
            DFHPC TYPE=LINK,                LINK TO THE                 X
                    PROGRAM=WINDOWS             WINDOWS PROGRAM
            SPACE
            DFHPC TYPE=RETURN               RETURN TO CICS
            END
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

LINKING FROM A COMMAND LEVEL PROGRAM

In a Command Level program prior to release 3 of CICS, you can not obtain a terminal I/O area, therefore the only available method is as described in method 2, above, passing the command in the Transaction Work Area. For version 3 of CICS, you can link to the WINDOWS program passing a TIOA as described in the previous example. Following is an example of the coding required using Command Level COBOL:

Command Level COBOL Example

```
IDENTIFICATION DIVISION.
PROGRAM-ID.    WNDOCALL.

DATA DIVISION.
WORKING-STORAGE SECTION.

LINKAGE SECTION.

01              DFHBLLDS.
                02      FILLER                  PIC  S9(8) COMP.

                02      TWA-POINTER             PIC  S9(8) COMP.

01              TWA-AREA.
                05      WNDO-COMMANDS           PIC  X(8).

    PROCEDURE DIVISION.

*
******          ESTABLISH ADDRESSABILITY TO TWA
*
                EXEC CICS ADDRESS TWA(TWA-POINTER) END-EXEC.


*
*****           LINK TO THE WINDOWS MODULE
*

                MOVE 'WNDO,ON' TO WNDO-COMMANDS.
                EXEC CICS LINK PROGRAM('WINDOWS') END-EXEC.

*
*****           RETURN TO CICS
*
                EXEC CICS RETURN END-EXEC.
```

[Note]: When defining your calling program to CICS, be sure to specify a TWASIZE of at least 32 bytes.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

BYPASSING THE CICS-WINDOWS USER CONFIGURATION DISPLAY

When activating or deactivating CICS-WINDOWS from a user program, it is often desirable to suppress the User Configuration Display that is normally produced in response to the WNDO,ON or WNDO,OFF command. Without doing this, if the calling program generates a terminal display after linking to WINDOWS or if control is passed to another program, such as the sign-on program, which creates a terminal display, the WINDOWS User Configuration Display will appear to the operator as an intermittent flash between displays.

To suppress the User Configuration Display, select the BYPASS STATUS SCREEN option in the User Option Table. See section 11 - *CUSTOMIZATION* for a more complete discussion.

With the BYPASS STATUS SCREEN option in effect, the User Configuration Display can still be invoked by entering the WNDO,INQ or the WNDO (no operand) command. It will not display in response to a WNDO,ON or WNDO,OFF command, whether invoked from a user program or from the terminal, nor will the Auto-Purge messages generated by the FORCE PURGE AT SIGNON AND SIGNOFF option display.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

BUILT-IN FUNCTIONS

There are five interface routines that can be executed by a direct call to CICS-WINDOWS. These interfaces are used primarily by a number of vendor packages in order to operate properly with CICS-WINDOWS, primarily to handle the pseudo IDs correctly. They can be used by any user program as long as the linkage standards are strictly adhered to. The built-in functions can only be called from an ASSEMBLER program.

The five built-in function routines are:

1) Test if CICS-WINDOWS is active on a terminal.
2) Purge CICS-WINDOWS from a terminal.
3) Obtain the real terminal ID from a pseudo ID.
4) Alter the transaction to be initiated in a recurring session.
5) Obtain addressability to the active user table.

OBTAINING THE BUILT-IN FUNCTION ENTRYPOINT

All of the built-in functions are accessed through a table of entrypoint addresses located at the top of the WINDOWS module. Thus, the technique for using a built-in function interface is to obtain the program entrypoint for WINDOWS, then index into this table to get the entrypoint for the desired routine, then perform a BALR or BASR to the function.

There are two methods of obtaining the program entrypoint for the WINDOWS module. Either can be done from a command-level program, but the second method requires addressability to the CSA, which is still possible in all releases of CICS.

Method 1: Perform a program LOAD. In either macro-level or command-level, you can issue a program control LOAD and obtain the entrypoint for program WINDOWS. If your calling program is not part of a CICS task that will go to normal transaction end when finished, you must do a DELETE (macro-level) or RELEASE (command-level) when finished to reduce the in-use count.

Method 2: You can go through the CSA to obtain the entrypoint for WINDOWS as well as determining if CICS-WINDOWS has been activated at all. The advantage of this method is it bypasses the CICS call which means the overhead is less. If your program is a global CICS exit with lots of activity, this method should be used. Note that the technique documented here is only valid for CICS-WINDOWS release 3.2 or above. Older releases must use the technique documented for that version, which will not work for release 3.2.

The coding for the direct-access method of obtaining the WINDOWS entrypoint is as follows:

```
L       R15,CSATCRWE            Load ZARQ address
CLI     CSACIREL,X'32'          CICS release 3.2 or greater?
BNL     *+8                             Br if yes
ICM     R15,8,=X'00'            Clear high-byte in address
SH      R15,=H'16'              Back-up 16 bytes
CLC     =CL8'WINDOWS',0(R15)    Is Windows hooked?
BNE     EXIT                    No, can't be active
L       R15,12(R15)             Load Windows entrypoint
```

You can eliminate the test for CICS release 3.2 and 3.3 if your calling program is not release independent. For CICS 3.1 and below, the high-order byte of the terminal control read-write routine con-

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

tains a non-zero value which must be cleared if your calling program could be running in 31-bit addressing mode.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

[Note]: For versions 3.2 and 3.3 of CICS, it is still possible to reference the CSA. For details on obtaining the CSA address, see PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3, later in this section.

THE BUILT-IN FUNCTION ENTRYPOINT LIST

Once you have the program entrypoint for WINDOWS, you must back-up 32 bytes to the beginning of the address list of built-in function entrypoints. The first five addresses in this list are the entrypoints for interfaces one through five. The next three words contain zeros. The five routines in the list are:

1). Test if CICS-WINDOWS is active on the terminal, return no other information.

2). Purge CICS-WINDOWS from the terminal, aborting any conversational tasks that may be inactive.

3). Obtain the real terminal ID from a pseudo ID.

4). Alter the next transaction to be automatically initiated in a recurring session.

5). Access the CICS-WINDOWS active user table for this terminal.

Each of these routines are described in detail following.

LINKAGE CONVENTIONS

Linkage to all five built-in functions is effectively the same, although interface number 4 requires more information to be passed. All five routines require the following register contents upon entry:

R1 - Contains the address of a 4-character terminal ID.
R13 - Contains the address of a 72-byte task savearea (Do not use the CSA).
R14 - Contains the return address.
R15 - Contains the built-in function entrypoint.

Upon return, all registers contain the same values as when the call was issued, except register 1 and sometimes register 15, as described in the individual routines following. Register 1 will always contain x'00000000' if CICS-WINDOWS is not active on the requested terminal. If it is active, register 1 will contain the information requested by the chosen interface, if applicable.

An example of the coding required to access built-in function number 3, obtaining the real terminal ID, is as follows, assuming R15 contains the entrypoint for the WINDOWS module.

```
SH      R15,=H'32'          Back-up to entrypoint list
L       R15,((3-1)*4)(R15)  Load third entrypoint
LA      R13,SAVEAREA        Point R13 to savearea
LA      R1,EIBTRMID         Point R1 to the terminal ID
BALR    R14,R15             Perform Interface 3
LTR     R1,R1               Is Windows active ?
BZ      EXIT                  Br if no
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In this example, R1 now contains the real terminal ID.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

To use any of the built-in functions, first obtain the program entrypoint for the WINDOWS module as described in *OBTAINING THE BUILT-IN FUNCTION ENTRYPOINT* above.

The following examples assume that address is in R15:

Built-in Function 1 – Test is CICS-WINDOWS is active on a terminal.

This interface simply returns a zero or non-zero value in R1. Zero means CICS-WINDOWS is not active on the terminal, non-zero means it is.

Coding required:

```
SH      R15,=H'32'          Back-up to entrypoint list
L       R15,((1-1)*4)(R15)  Load first entrypoint
LA      R13,SAVEAREA        Point R13 to savearea
LA      R1,EIBTRMID         Point R1 to the terminal ID
BALR    R14,R15             Perform Interface 1
LTR     R1,R1               Is Windows active ?
BZ      EXIT                  Br if no
```

Built-in function 2 – Purge CICS-WINDOWS from a terminal.

This interface will return a zero value in R1 if CICS-WINDOWS is not active on the terminal. Otherwise an internal PURGE commands is issued, which deactivates CICS-WINDOWS from the terminal after first terminating any conversational tasks that may have been toggled out-of. The current session is not disturbed and the terminal ID is reset to the real ID.

Coding required:

```
SH      R15,=H'32'          Back-up to entrypoint list
L       R15,((2-1)*4)(R15)  Load second entrypoint
LA      R13,SAVEAREA        Point R13 to savearea
LA      R1,EIBTRMID         Point R1 to the terminal ID
BALR    R14,R15             Perform Interface 2
```

[Note] Built-in Function 2 cannot be used if the calling terminal is running as an asynchronous task. That is, the calling task must be attached to a TCTTE. Since the Node Error Program runs asynchronously, this function cannot be called from DFHZNEP. See the following discussion, entitled *DEACTIVATING CICS-WINDOWS WHEN A NODE ERROR OCCURS*, for details of the procedure to be used in this case.

[Note] It is possible to use Built-in Function 2 to purge a terminal other than the one the calling task is attached to. In this example, R1 points to EIBTRMID, but it may point to any 4-byte field containing a valid terminal ID.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Built-in function 3 – Obtain the real terminal ID from a pseudo ID.

This interface will return a zero value in R1 if CICS-WINDOWS is not active on the terminal. Otherwise, R1 contains the real terminal ID (the actual ID, not the address of the ID). In addition, R15 contains the following values .

BYTE 1	-	Binary zero
BYTE 2	-	C'Y' if pseudo IDs are in use
BYTE 3	-	Binary value from 1 to 9 representing the number of virtual terminals in use
BYTE 4	-	Binary value from 1 to 9 representing the current session number.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Coding required:

```
SH      R15,=H'32'          Back-up to entrypoint list
L       R15,((3-1)*4)(R15)  Load third entrypoint
LA      R13,SAVEAREA        Point R13 to savearea.
LA      R1,EIBTRMID         Point R1 to the terminal ID.
BALR    R14,R15             Perform Interface 3
LTR     R1,R1               Is Windows active ?
BZ      EXIT                Br if no
PRINT \p  PAGE "
```

Built-in function 4 - Alter the transaction in a recurring session.

Built-in function 4 – Alter the transaction in a recurring session.

This is a special-purpose interface used in application integration. It can be used to make a decision, while in one session, as to what transaction to initiate in another session usg a recurring application start-up table. In most cases, the same functionality can be obtained by accessing the active user table with interface 5, determining the pseudo ID of the desired session and issuing a START for a transaction in that session. When the operator subsequently toggles to the session, the started transaction will initiate.

The problem with using START, however, is that it may be possible to issue several START commands before a toggle is done into the target session. In other words, the user program must keep track of what STARTS have been issued for which sessions to avoid unnecessary queuing up of several pending transactions.

By designating a recurring session, one which has an associated transaction which is to initiate every time the session is entered, CICS-WINDOWS will do the start or attach only when the session is activated. With interface 4, you can alter both the transaction and the data to be passed to the transaction based on some criteria in the current session.

For details of the coding involved, see *CHANGING THE TRANSACTION CODE IN A RECURRING SESSION*, later in this section.

Built-in function 5 – Obtain addressability to the active user table.

This interface will return a zero value in R1 if CICS-WINDOWS is not active on the terminal. Otherwise, R1 contains the real address of the CICS-WINDOWS active user table entry for that terminal. In addition, R15 contains the same information returned in interface 3, as follows:

BYTE 1 -	Binary zero
BYTE 2 -	C'Y' if pseudo IDs are in use.
BYTE 3 -	Binary value from 1 to 9 representing the number of virtual terminals in use.
BYTE 4 -	Binary value from 1 to 9 representing the current session number.

The active user table is a table which is maintained of all terminal operators who have activated CICS-WINDOWS on their terminal. Much of the information in the active user table is useful when you are modifying application programs to run in a more integrated fashion in a windowing environment. For details of the format and use of the active user table, see *ACCESSING OR ALTERING THE ACTIVE USER TABLE ENTRY* later in this section.

Coding required:

```
SH      R15,=H'32'          Back-up to entrypoint list
L       R15,((5-1)*4)(R15)  Load fifth entrypoint
LA      R13,SAVEAREA        Point R13 to savearea
LA      R1,EIBTRMID         Point R1 to the terminal ID
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
BALR    R14,R15          Perform Interface 5
LTR     R1,R1            Is Windows active ?
BZ      EXIT               Br if no
USING   WINTABLE,R1      Address the active user table
```

USES FOR THE BUILT-IN FUNCTIONS

There are many uses of the built-in functions described above. Some are listed in the following examples along with the coding required:

DEACTIVATING CICS-WINDOWS WHEN A NODE ERROR OCCURS

The node error program in CICS (DFHZNEP) controls the action to be taken when a terminal error occurs. The *CICS CUSTOMIZATION GUIDE* contains instructions for modifying this program to conform to your environmental specifications.

It is good practice, and sometimes critically important, to add some logic to DFHZNEP to deactivate CICS-WINDOWS at a terminal any time a terminal error occurs where DFHZNEP allows or forces the terminal operator to be logged off. Failure to deactivate CICS-WINDOWS when an error resulting in a log-off occurs can produce the following unsatisfactory conditions:

1) There is a potential security breach here. If the terminal is logged off because of an error, CICS-WINDOWS does not have a chance to go through its session termination logic, therefore the session termination options are not performed. If another operator logs on to the terminal and presses the toggle key, they will continue in whatever transaction was in progress in that virtual terminal prior to the terminal error occurring. This may be a transaction that they should not be authorized to use.

   Note that this condition can also be controlled with the FORCE PURGE AT SIGNON AND SIGNOFF option, but many people prefer the technique of deactivating CICS-WINDOWS from DFHZNEP, in order to clean everything up when the error occurs, rather than waiting till the next signon.

2) With some security packages, this condition can result in a storage violation in CICS. This is because CICS-WINDOWS has saved an address pointer from the TCT which points to a security system control block. When the forced log-off occurs, the security package releases its control blocks for that terminal. Upon re-entering a previously saved session in CICS-WINDOWS, the old address of the control block is restored in the TCT, which usually will no longer point to the same or even a valid security control block.

   Security packages that are known to result in this situation are ACF2, RACF and TOP-SECRET. As a general rule, if you are using any package other than normal (non-RACF) CICS security, you should install this interface.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CODING THE DFHZNEP INTERFACE

There are two sample programs in the source file of the installation tape called WNDONEPC and WNDONEPM. WNDONEPC is a command level DFHZNEP example, and WNDONEPM is macro-level. You may use the appropriate program or simply insert the necessary code in your DFHZNEP program.

Locate the point in DFHZNEP where the terminal is to be disconnected. At this point, or anywhere in the main logic of DFHZNEP, insert the following instructions:

Note that you should test for the call to DFHZNEP that occurs at session start-up and skip the purge call. Otherwise you can have problems at log-on time.

CODING REQUIRED FOR COMMAND-LEVEL DFHZNEP.

```
            CLI     TWAEC,TCZOPSIN          Session just opened?
            BE      NEP0400                 Yes, skip purge
            EXEC    CICS,HANDLE,CONDITION,PGMIDERR(NEP0400).

            EXEC    CICS,LINK,PROGRAM('WNDOZNEP'),COMMAREA(TWATCTA).
NEP0400     EQU     *
            .
            .
            DFHEISTG
            DFHEIEND
            END     DFHZNENA                See note below
```

[Note]: In the CICS 3.2 and 3.3 default version of DFHZNEP, which is called DFHZNEPX, it was necessary to add this label to the END statement for the linkage editor to compute the correct entrypoint to the program. Check your link-edit output - the entrypoint should be at offset 48.

CODING REQUIRED FOR MACRO-LEVEL DFHZNEP.

```
            CLI     TWAEC,TCZOPSIN          Session just opened?
            BE      NEP0400                   Yes, skip purge
            DFHPC TYPE=LINK,PROGRAM=WNDOZNEP,COND=YES  PURGE  WNDOZNEP
NEP0400     EQU     *
```

DEACTIVATING PRIOR TO AUTOMATIC LOG-OFF OR SIGN-OFF

Many security packages and VTAM session managers can perform an automatic log-off or automatic sign-off at a terminal after a specified period of inactivity. Or, you may have an in-house program or CICS modification which performs this activity. It is usually desirable to deactivate CICS-WINDOWS when this occurs.

In many cases, the action taken to force the terminal to log-off generates a node error in CICS and the DFHZNEP program is invoked. If this is the case, the interface logic described above in *DEACTIVATING CICS-WINDOWS WHEN A NODE ERROR OCCURS* will suffice. If there is any question, a CICS auxiliary trace of the automatic log-off will show you if DFHZNEP is getting invoked.

If DFHZNEP is not getting invoked, your security package or session manager may have a user exit point where you could place this interface code. If that is not the case, and you feel that it is necessary to handle this condition, please notify both Unicom Systems and the vendor of the product in question. There are several vendor packages on the market which currently interface to CICS-WINDOWS for one reason or

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

another, and most vendors are quite willing to insert this interface logic into their product. Two security packages which currently call CICS-WINDOWS prior to performing an automatic log-off are ALERT-CICS, and CA-SENTINAL.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

OBTAINING THE REAL TERMINAL ID FROM A PSEUDO ID

When pseudo terminal IDs are in use, it is often desirable to obtain the real terminal ID, when all that is known is the pseudo ID.

Interface 3 is currently used by several vendor packages which need to maintain information about the terminal by the real ID, and cannot process a pseudo ID. Among these products are DATAPACKER, SUPER-OPTIMIZER, ALERT-CICS, and CTOP-III. If you are using ACF2, there is a user exit available in it called the TERMINAL IDENTIFICATION EXIT, into which this interface can be inserted. For release 5.0 of ACF2 and above, this interface is necessary for proper operation.

CODING THE ACF2 TERMINAL ID RETRIEVAL EXIT

For ACF2 users, there is a sample program in the source file of the installation tape called WNDOACF2. You may use this program as your terminal identification exit or simply insert the necessary code in your program.

The interface code for this routine is exactly the same as interface 3, described BUILT-IN FUNCTIONS, above. Simply load the address of the known terminal ID from wherever it exists (EIBTRMID if this is a command level program) into register 1 and call the WINDOWS program as illustrated. The real terminal ID will be returned in register 1.

ACCESSING OR ALTERING THE ACTIVE USER TABLE ENTRY

The active user table is a table that is maintained by CICS-WINDOWS of all terminal operators who have activated CICS-WINDOWS on their terminal. Much of the information in the active user table is useful when you are modifying application programs to run in a more integrated fashion in a windowing environment.

There is a general purpose interface routine that allows you to call the WINDOWS program and get addressability to the active user table entry for the current terminal in use. Once addressability is obtained, you can use any of the information in the table to make decisions in your program, or in some cases, you may modify certain information to cause CICS-WINDOWS to act differently.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE ACTIVE USER TABLE

The exact format and contents of the active user table entry are contained in the source module called WNDOINT5, which is distributed on the installation tape.

In general, the active user table contains the following information:

1) The real terminal ID and all pseudo IDs in use.
2) The aid byte of the toggle-forward key.
3) The aid byte of the toggle-backward key.
4) The aid byte of the window key.
5) The aid bytes of all direct session keys
6) The aid bytes of the window-switch keys.
7) The aid bytes of the scrolling keys.
8) The total number of sessions allocated to the terminal.
9) The current session number.
10) The number of windows allocated.
11) The current transaction in progress in each session.
12) The current program in progress in each session.
13) The cursor position of the last terminal input message.

[Note]: The Active User Table Entry is located above the 16M line; therefore, your program must be operating in 31-bit addressing mode to access it.

CODING THE ACTIVE USER TABLE ACCESS INTERFACE

There is a sample program in the source file of the installation tape called WNDOINT5. You may use this routine to insert the necessary code in your program:

```
        EXEC    CICS,LOAD,PROGRAM('WINDOWS'),
                ENTRY(R15).             Get Windows entrypoint
        SH      R15,=H'32'              Back-up to entrypoint list
        L       R15,((5-1)*4)(R15)      Load fifth entrypoint
        LR      R2,R13                  Save current R13
        LA      R13,SAVEAREA            Point R13 to savearea
        L       R1,TWATCTA              Load TCTTE address
        BALR    R14,R15                 Perform Interface 5
        LR      R13,R2                  Restore R13
EXIT    EQU     *
        .
        .
        .
        DFHEISTG
SAVEAREA        DS                      18F
*
*  Upon return, R1 contains the address of the active user table entry
*
*  And R15 contains the following values ...
*       BYTE 1  -   Binary zero
*       BYTE 2  -   C'Y' if pseudo IDs are in use.
*       BYTE 3  -   Binary value from 1 to 9 representing the number of virtual terminals in use.
*       BYTE 4  -   Binary value from 1 to 9 representing the current session number.
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

USES FOR ACCESSING THE ACTIVE USER TABLE

Some of the possible uses for obtaining the information in the active user table might be as follows:

1) Obtaining the pseudo ID of another session in order to start a transaction destined for that session.

2) Determining what transaction is active in another session in order to allow or disallow certain activity in this session.

3) Using the last input cursor position to "pick" an item from the incoming terminal I/O area, to use as the key of a transaction to be routed to another session.

ALTERING THE ACTIVE USER TABLE ENTRY

You can change some of the information in the active user table under program control. However, you must be very careful not to change anything else, as doing so could cause an abnormal termination or other unpredictable results. The fields that may be changed are:

1) The aid byte of the toggle-forward key.
2) The aid byte of the toggle-backward key.
3) The aid byte of the window key.
4) The aid bytes of all direct session keys
5) The aid bytes of the window-switch keys.
6) The aid bytes of the scrolling keys.
7) The number of rows and/or columns to shift when one of the scrolling keys is pressed.
8) The total number of sessions allocated. (You can decrease this but not increase it).

You can disable the function of one or more of the hot keys by moving a x'00' to the corresponding aid byte for that key. This might be useful if you wanted to control the point at which you will allow an operator to toggle out of an application program.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CHANGING THE TRANSACTION CODE IN A RECURRING SESSION

Transaction start tables (Application Start-up Tables in the WAUX transaction) are normally used to initiate a transaction in a session upon first toggling into that session. If RECUR is specified, however, the transaction will be initiated every time the session is entered via a toggle or window-switch function. This feature allows you to customize and integrate your applications by causing things to happen automatically when a session is entered.

It is sometimes desirable to alter the transaction to be initiated in a recurring session, based upon some logic of a control program running in another session. This can be accomplished with one of the built-in functions, interface number 4. To use this feature, do the following:

1). Define an application start-up table connected to the profile to be used.

2). For the session or sessions to be altered, designate the transaction code to be started first. This can be a dummy trancode if desired, as your program can set the correct one.

3). Code RECUR on all sessions where applicable.

4). If you intend to pass data to the transaction that will change with each invocation, fill the DATA field with dummy information. Completely fill the field so there will be room to alter it with data of different lengths.

5). In your control program which makes the decision as to what transaction should be initiated next in this session, define a parameter list as described below, then call the WINDOWS program using built-in function interface 4, passing the address of the parameter list.

6). This causes WINDOWS to alter the application start-up entry for the designated session with the information passed in the parmlist. The next time the operator toggles into this session, the new transaction will be initiated with any data to be passed to it.

CODING THE AUTO-START TRANSACTION-CHANGE INTERFACE

The following instructions illustrate this procedure:

```
*
*****  Definition of parameter list to pass with interfa
e 4
*
BIF4PARM     DS      0F
BIF4TERM     DS      CL4     Terminal ID
BIF4SESS     DS      XL1     Session number to be altered
BIF4TRAN     DS      CL4     New transaction code to initiate
BIF4METH     DS      CL1     Method of initiation ...
*                                    A = Attach
*                                    S = Start
BIF4DLEN     DS      XL1     Length of data to be passed ...
*                                    x'00' = none
*                                    x'FF' = leave as i          X'01'-X'28' (data length)

BIF4DATA     DS      CL56    Data to be passed to transaction, if any
                     .
                     .
                     .
 *
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
*****    Set up parameter list to transaction to be invoked
*
         MVC       BIF4TERM,EIBTRMID
         MVC       BIF4SESS,...
         MVC       BIF4TRAN,...
         MVC       BIF4METH,...
         MVC       BIF4DATA,...
                   .
                   .
                   .
*
*****    Required coding to call interface 4
*
         EXEC      CICS,LOAD,PROGRAM('WINDOWS'),                X
               ENTRY(R15).                    Get Windows entrypoint
         .
         .
         or ...
         .
         .
         L         R15,CSATCRWE             Load ZARQ address
         CLI       CSACIREL,X'32'           CICS release 3.2 or greater?
         BNL       *+8                            Br if yes
         ICM       R15,8,=X'00'             Clear high-byte in address
         SH        R15,=H'16'               Back-up 16 bytes
         CLC       =CL8'WINDOWS',0(R15)     Is Windows hooked?
         BNE       EXIT                     No, can't be active
         L         R15,12(R15)              Load Windows entrypoint
         .
         .
         .
         SH        R15,=H'32'               Back-up to entrypoint list
         L         R15,((4-1)*4)(R15)       Load fourth entrypoint
         LA        R13,SAVEAREA             Point R13 to savearea
         LA        R1,BIF4PARM              Load parmlist address
         BALR      R14,R15                  Perform Interface 4
         LTR       R1,R1
         BZ        EXIT                     Exit if Windows not active
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SAMPLE PROGRAMS ON THE INSTALLATION TAPE

Several programs and routines are distributed on file number one of the installation tape. For DOS/VSE users, these programs are cataloged as sub-library type 'Z'.

Each program is written in Assembler language. The on-line programs and routines use macro-level CICS. Each program contains documentation at the beginning of the source member that describes its function and usage.

To make use of any of these programs, move the desired member to your editor library to make any desired modifications, then assemble and link-edit the member into the desired core-image/load library.

The programs and routines on the tape are:

MENUEXIT — This is a command-level example user exit for the Menu Generation Feature. This exit may be used to disallow certain options of the menu according to any logic you desire. For more information, see *THE MENU GENERATION FEATURE USER EXIT*, earlier in this section.

WNDOACF2 — This is a complete terminal identification exit that can be used with ACF2, release 5.0 or greater. ACF2 requires this exit to be installed if pseudo terminal IDs are in use in CICS-WINDOWS. The use of this exit routine causes CICS-WINDOWS to pass the real terminal ID to ACF2 each time it needs to reference the current terminal identification.

To install, assemble and link this routine, then activate the terminal identification exit in ACF2. Or you can insert the necessary interface instructions at the terminal identification exit point in the exit program skeleton module distributed with ACF2.

WNDOCSSF — This program contains logic to purge CICS-WINDOWS from a terminal any time a sign-off transaction is performed. This approach could be used instead of using the REQUIRE WNDO,OFFAT LOGOFF or FORCE PURGE AT SIGNON AND SIGNOFF option to ensure that CICS-WINDOWS is inactive when the operator signs off.

WNDOCSSF first links to WINDOWS with a purge command for the terminal, then restores the original input terminal I/O area and exits to DFHSNP. DFHSNP will then return to CICS.

To install this program, assemble and link it to the CICS execution library, then change the CSSF transaction code (or any other sign-off transaction that you use) in the PCT to point to this program instead of DFHSNP.

WNDOCSSN — This program contains logic to purge CICS-WINDOWS from a terminal any time a user signs on to CICS. This approach could be used instead of using the FORCE PURGE AT SIGNON AND SIGNOFF option to ensure that CICS-WINDOWS is inactive on a terminal each time a sign-on occurs.

In its current form, WNDOCSSN first links to WINDOWS with a purge command for the terminal, then restores the original input terminal I/O area and link to DFHSNP. Upon return from the sign-on program, another link is made to CICS-WINDOWS with an ON command to reactivate CICS-WINDOWS. If you do not want to automatically activate CICS-WINDOWS, remove the final link to WINDOWS and let it simply return to CICS.

To install this program, assemble and link it to the CICS execution library, then change the CSSN transaction code (or any other sign-on transaction that you use) in the PCT to point to this program instead of DFHSNP.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDOINT3 This is a sample of the interface routine that can be used to retrieve session information from CICS-WINDOWS. The information that can be retrieved is the current session number, the real terminal ID, CICS-WINDOWS status, the number of virtual terminals and whether pseudo IDs are in use or not.

This sample program cannot be used by itself. It contains the necessary instructions to call CICS-WINDOWS and obtain the information returned. To use, insert these instructions into your program and assemble.

WNDOINT5 This is a sample of the interface routine that can be used to access the active user table entry for a terminal.

This sample program cannot be used by itself. It contains the necessary instructions to call CICS-WINDOWS and obtain the address of the active user table. It also contains the format of the active user table. To use, insert these instruction into your program and assemble.

WNDONEPC This is a command-level version of DFHZNEP containing the interface logic to deactivate CICS-WINDOWS when a node error occurs. If you have made no other modification to the system-supplied version of DFHZNEP, you can use this program as it is. Otherwise, you should extract the instructions for the CICS-WINDOWS interface routine and insert it into your version of DFHZNEP.

To install, assemble and link-edit to the CICS execution library, then cycle CICS.

WNDONEPM This is a macro-level version of DFHZNEP containing the interface logic to deactivate CICS-WINDOWS when a node error occurs. If you have made no other modifications to the system-supplied version of DFHZNEP, you can use this program as it is. Otherwise, you should extract the instructions for the CICS-WINDOWS interface routine and insert it into your version of DFHZNEP.

To install, assemble and link-edit to the CICS execution library, then cycle CICS.

WNDOOFFX This is a sample Terminal Deactivation Exit which gets control when CICS-WINDOWS is about to be deactivated at a terminal.

WNDOPURG This program contains logic to purge CICS-WINDOWS from a terminal when linked-to from a user program.

There may be occasions in your environment when it is desirable to deactivate CICS-WINDOWS at a terminal before some event occurs. Perhaps you have a routine to automatically sign-off after a specified period of inactivity. In order to do a complete sign-off, you should also deactivate CICS-WINDOWS, if it is active. This program can be used for that.

To install this program, assemble and link it to the CICS execution library, then define it in the PPT. You can then issue a macro level or command level LINK command any time you want to deactivate CICS-WINDOWS. Note that this program will only deactivate the terminal currently attached to the task. If you want to purge a different terminal, you could add logic to pass a terminal ID to WNDOPURG.

WNDORSDD (DOS/VSE version)
WNDORSDM (MVS version)
This is the DOS/VSE and MVS version of a batch program that can be used to delete any pseudo terminal ID entries from the CICS restart dataset prior to activating CICS with the emergency restart option.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

In CICS 1/7 and above, the TCTTE entries are logged to the restart dataset each time a terminal logs on. This includes the entries containing a pseudo terminal ID, if in use. Then when an emergency restart is done, separate TCTTE entries are created for each pseudo terminal ID, which is not correct. If autoinstall is used for TCT definitions, these pseudo entries are automatically deleted after a few seconds. While this condition does not really cause any problems with the operation of CICS, it means that excess storage is tied up for TCTTE entries that are never used.

To alleviate this problem, this batch program can be included in the CICS start-up JCL, prior to initializing CICS. It runs very quickly, and will scan through the restart dataset (DFHRDS) deleting all pseudo terminal entries that it finds. Then when an emergency restart is done, only the real terminal entries are created.

PERFORMING MACRO LEVEL FUNCTIONS IN CICS 3.2 AND 3.3

Despite IBM's general instruction, macro level assembler programs can still operate in CICS 3.2, 3.3 and 4.1, with certain restriction. The following rules apply:

1) No file control activity. The DFHFC macros and all associated control blocks have been removed. File I/O must be performed in command level.

2) No GETMAIN CLASS=TEMPSTRG. You must obtain USER storage for DFHTS functions.

3) Must be in 31-bit addressing mode. Most control blocks and nucleus modules are located above the line.

ACCESSING THE TCA AND CSA

The following instructions will locate the TCA in MVS/XA or MVS/ESA:

```
L       R13,X'021C'(R0)     Load TCA address
L       R13,X'D0'(R13)      Locate the CSA
L       R13,X'14'(R13)      Locate the CSA
L       R13,8(R13)          Locate the CSA
USING   DFHCSADS,R13        Address the CSA
L       R12,CSACDTA         Load current TCA address
USING   DFHTCADS,R12        Address it
```

[Note]: Source copybooks and macro-level macros are located in CICS321.SDFHMAC.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Section 16. MESSAGES

AUXILIARY FUNCTIONS MESSAGES "WA" PREFIX

Messages produced by the Auxiliary functions transaction are prefixed by "WA" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WA00001. CICS-WINDOWS PASSWORD ERROR, CODE=x

This message indicates that the product has expired or cannot be used because of a password error or installation error. The product is now inactive and cannot be used.

The "x" in the above message will be one of the following error codes:

1 - The current date is greater than the password expiration date.
2 - Module "STSPASS" could not be found in a library in the search string.
3 - The product ID could not be located in STSPASS.
4 - The password in STSPASS is invalid.
5 - Module "STS0100" could not be found in a library in the search string.
6 - STSPASS contains a CPU ID that is invalid for the machine that it is installed on.
7 - The password entered is not valid for the requested product.
9 - An undetermined error has been encountered. Please call Unicom Systems Technical Support at (818) 838-0606.

ACTION: Correct the error if possible, or call Unicom Systems Technical Support at (818) 838-0606.

WA00002. FILE "xxxxxxxx" IS NOT OPEN.

A file defined to CICS-WINDOWS (in the Option table) is not available for the requested function.

ACTION: If the function is to be performed, the file must be made available.

WA00003. FILE "xxxxxxxx" HAS NOT BEEN DEFINED.

The file denoted by xxxxxxxx is not defined in the CICS FCT.

ACTION: Ensure that the file name is spelled correctly and that the file is available to CICS-WINDOWS.

WA00004. SCREEN "xxxxx" IS NOT ON FILE.

The WNDOFIL file contains records that are for CICS-WINDOWS internal use. Amongst these are records that are used to map some of the CICS-WINDOWS screen displays. This message notifies that CICS-WINDOWS was unable to locate a necessary screen record.

ACTION: All necessary screen records are shipped with the product and are pulled from the installation tape during the installation process. If a screen record has been deleted, you may run the installation step that initializes the WNDOFIL VSAM file with the MODE=REINSTALL keyword.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WA00005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-WINDOWS could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WA00030. PCT ENTRY FOR TRANSACTION xxxx MUST HAVE A TWASIZE OF AT LEAST 32 BYTES

The PCT entry for the transaction indicated by xxxx must have a TWA size of at least 32 bytes.

ACTION: Using RDO or the DFHPCT macro, increase the TWA size.

WA01003. WINDOWS IS NOT AVAILABLE.

CICS-WINDOWS could not find the program "WINDOWS". Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WA02001. INVALID SELECTION. PLEASE SELECT AGAIN.

The selection was invalid.

ACTION: Review the valid selections in the Technical Reference Guide, then respond accordingly.

WA02004. VERSION 3.0 USER OPTION TABLE HAS BEEN CREATED.

This message is issued in response to a dynamic creation of the user option table.

ACTION: No action is required.

WA06001. MAKE CHANGE AND PRESS "ENTER" TO UPDATE THE USER OPTION TABLE.

The dynamic User Option table is being viewed.

ACTION: If you wish to change the Option table, change the appropriate field and press ENTER.

WA06002. USER OPTION TABLE HAS BEEN CHANGED.

This message is merely to confirm that a requested change to the User Option table was completed successfully.

ACTION: No action is required.

WA06003. USER OPTION TABLE IS BEING MODIFIED ON ANOTHER TERMINAL

A request to edit the User Option table has failed because the table is being edited on another terminal.

ACTION: Wait until the other user has finished editing the table.

WA07001. PROFILE ID "xxxx" ALREADY EXISTS.

An attempt to create a Profile with the ID of xxxx has failed because another Profile with the same ID already exists.

ACTION: Choose another ID for the new Profile.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WA07002. OPTION TABLE HAS CHANGED SINCE PREVIOUS INPUT. CHANGES IGNORED.

Steps have been taken to prevent more than one person from changing customization options at the same time; however it is possible to defeat this safety measure. If CICS-WINDOWS detects a change in this table (that did not originate from this terminal) while this terminal was editing this table, this message will be displayed.

ACTION: Exit from this table to the main menu, and reenter this table. Then perform the desired changes.

WA07004. THE FIRST PROFILE TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the Profile entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA07005. THE LAST PROFILE TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the Profile entries, the last entry on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WA07006. PROFILE "xxxx" HAS BEEN DELETED.

In response to a request to delete a Profile record, the record was successfully deleted.

ACTION: No action is required.

WA07007. "xxxx" IS THE DEFAULT PROFILE. IT CANNOT BE DELET
D.

In response to a request to delete a Profile record, the request failed because the Default Profile may not be deleted.

ACTION: No action is required.

WA07008. PROFILE "xxxx" HAS BEEN ADDED.

This message merely confirms that the Profile xxxx was successfully added to the table.

ACTION: No action is required.

WA07009. PROFILE "xxxx" HAS BEEN CHANGED.

This message merely confirms that the Profile xxxx was successfully altered.

ACTION: No action is required.

WA07010. PROFILE "xxxx" WAS NOT FOUND.

In response to a Find function, CICS-WINDOWS was unable to locate the record in the User Option table being searched.

ACTION: If multiple files are being used for CICS-WINDOWS, then the record may exist in another file.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WA07011. PROFILE "xxxx" IS REFERENCED BY AN AUTO-INIT ENTRY. IT CANNOT BE DELETED

Deletion of the Profile xxxx failed because the Profile is referenced by at least one Auto-Init record.

ACTION: Delete or modify all Auto-Init entries that point to this profile. Then this profile may be deleted.

WA07012. PROFILE ID IS REQUIRED.

While adding a new Profile , the record cannot be added until a new Profile ID is entered.

ACTION: Key a valid profile ID and press ENTER.

WA08004. THE FIRST AUTO-INIT TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the Auto-Init entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA08005. THE LAST AUTO-INIT TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the Auto-Init entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA08009. AUTO-INIT TABLE HAS BEEN CHANGED.

This message merely confirms that the Auto-Init table was successfully altered.

ACTION: No action is required.

WA09001. APPL START ID "xxxxxxxx" ALREADY EXISTS.

An attempt to create an Application Start entry with the ID of xxxxxxxx has failed because another profile with the same ID already exists.

ACTION: Choose another ID for the new Application Start entry.

WA09004. THE FIRST APPLICATION START TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the Application Start entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA09005. THE LAST APPLICATION START TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the Application Start entries, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WA09006. APPLICATION START TABLE "xxxx" HAS BEEN DELETED.

In response to a request to delete an Application Start record, the record was successfully deleted.

ACTION: No action is required.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WA09008. APPLICATION START TABLE "xxxxxxxx" HAS BEEN ADDED.

This message merely confirms that the Application Start record xxxxxxxx was successfully added to the table.

ACTION: No action is required.

WA09009. APPLICATION START TABLE "xxxxxxxx" HAS BEEN CHANGED.

This message merely confirms that the Application Start record xxxxxxxx was successfully altered.

ACTION: No action is required.

WA09010. APPLICATION START TABLE "xxxx" WAS NOT FOUND.

In response to a Find function, CICS-WINDOWS was unable to locate the record in the User Option table being searched.

ACTION: If multiple files are being used for CICS-WINDOWS then the record may exist in another file.

WA09011. APPL START ID "xxxxxxxx" IS REFERENCED BY THE PROFILE TABLE. CAN'T DELETE.

Deletion of the Application Start record xxxxxxxx failed because the Application Start entry is referenced by at least one Profile record.

ACTION: Delete or modify all Profile entries so that no Profiles point to this Application Start entry. Then this record may be deleted.

WA09012. ID IS REQUIRED.

While adding a new record, the record cannot be added until a new ID is entered.

ACTION: Key a valid ID and press ENTER.

WA10004. THE FIRST TABLE ENTRY HAS BEEN DISPLAYED.

While browsing backward through the table, the first entry on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WA10005. THE LAST TABLE ENTRY HAS BEEN DISPLAYED.

While browsing forward through the table, the last entry on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WA10009. TABLE HAS BEEN CHANGED.

This message merely confirms that the table was successfully altered.

ACTION: No action is required.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WA11004. AT LEAST ONE FILE TABLE ENTRY IS REQUIRED..

The File table specifies the file that CICS-WINDOWS uses for functions such as mapping the screens of several CICS-WINDOWS screen displays.

ACTION: CICS-WINDOWS must have at least one file.

WA11009. FILE TABLE HAS BEEN CHANGED.

This message merely confirms that the File table was successfully altered.

ACTION: No action is required.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MESSAGE BROADCASTING MESSAGES "WB" PREFIX

Messages produced by WNDOMSG (the Message Broadcasting facility) are prefixed by "WB" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WB01005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-WINDOWS could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WB02004. THE FIRST MESSAGE DEFINITION HAS BEEN DISPLAYED.

While browsing backward through the message definitions, the first definition on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WB02005. THE LAST MESSAGE DEFINITION HAS BEEN DISPLAYED.

While browsing forward through the message definitions, the last definition on file has been displayed. This message also appears on the Status Display if no active messages are found.

ACTION: To browse through more entries you must browse in a backward direction.

WB02008. MESSAGE "xxxxxxxx" HAS BEEN ADDED.

This message merely confirms that the message definition xxxxxxxx was successfully added to the file.

ACTION: No action is required.

WB02009. MESSAGE "xxxxxxxx" HAS BEEN CHANGED.

This message merely confirms that the message definition xxxxxxxx was successfully altered.

ACTION: No action is required.

WB02010. MESSAGE "xxxxxxxx" WAS NOT FOUND.

In response to a Find function, CICS-WINDOWS was unable to locate the message definition xxxxxxxx in the file being searched.

ACTION: If multiple files are being used for CICS-WINDOWS then the definition may exist in another file.

WB04001. MESSAGE ID "xxxxxxxx" HAS NOT BEEN DEFINED.

The attempted function failed because the message definition could not be located in the file being searched.

ACTION: If multiple message files are in use, the message definition may reside in another file. In which case, you should ensure that the file is defined to CICS-WINDOWS. For more information see section 11 -

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CUSTOMIZATION, under *THE FILE TABLE*. If multiple message files are not in use, then the message definition does not exist.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WB04002. MESSAGE "xxxxxxxx" HAS BEEN SENT.

This message is issued in response to a Send command and confirms that the message has been successfully sent.

ACTION: No action is required.

WB04003. THERE IS NO TEXT FOR MESSAGE "xxxxxxxx"

An attempt to send a message failed because the text record for the message was not found.

ACTION: Create the message.

WB04004. MESSAGE "xxxxxxxx" IS CURRENTLY ACTIVE.

An attempt to send a message failed because the message has been previously sent and not purged, so that the message is still active.

ACTION: The message must be purged before it may be sent again.

WB04005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-WINDOWS could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WB04006. NO AVAILABLE TERMINALS WERE FOUND FOR MESSAGE "xxxxxxxx".

An attempt to send a message failed because CICS-WINDOWS did not find any terminals as specified in the message destination fields of the message definition, or because CICS-WINDOWS has not been initialized in this CICS.

ACTION: Review section 09 - MESSAGE BROADCASTING under FIELDS OF THE MESSAGE DEFINITION DISPLAY.

WB05001. MESSAGE ID "xxxxxxxx" HAS BEEN PURGED.

This message should not occur, if it does, it is due to an improper deletion of a message broadcast temp storage record.

ACTION: No action is required. The message cannot be retrieved.

WB05003. THERE ARE NO OUTSTANDING MESSAGES.

This message appears upon an attempt to view one's messages, if there are no messages that have been sent the user.

ACTION: No action is required.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WB05004. CICS-WINDOWS CANNOT RUN ON THIS TERMINAL.

A message has been broadcast to a terminal (or user) on which CICS-WINDOWS cannot be activated, due to the coding of the customization table (usually the terminal exclusion table). In order to deliver the message, CICS-WINDOWS must be active or be able to automatically activate on the terminal.

ACTION: If this terminal (or user) is to be eligible to receive message broadcasts, the terminal (or user) must also be able to activate CICS-WINDOWS.

WB08001."xxxxxxxx" IS NOT AN ACTIVE MESSAGE.

Upon attempting to display the Terminal Status of a message definition, CICS-WINDOWS noticed that the message was not active, so this message was issued to inform that there are not any terminals to display.

ACTION: No action is required.

WB08004. THE FIRST TERMINAL HAS BEEN DISPLAYED.

While browsing backward, the first terminal was reached.

ACTION: To browse through more entries you must browse in a forward direction.

WB08005. THE LAST TERMINAL HAS BEEN DISPLAYED.

While browsing forward, the last terminal was reached.

ACTION: To browse through more entries you must browse in a backward direction.

WB09006. MESSAGE "xxxxxxxx" HAS BEEN DELETED.

This message merely confirms that the message definition xxxxxxxx was successfully deleted from the file.

ACTION: No action is required.

WB09007. KEY DATA AND PRESS "ENTER" TO ADD MESSAGE.

In response to New function (creating a new message definition), this message is prompting you to key the desired data then press ENTER. The message will then be added.

ACTION: Key desired data then press ENTER.

WB09011. MESSAGE ID "xxxxxxxx" ALREADY EXISTS.

A request to add a new message definition failed because an old definition with the same message ID already exists.

ACTION: Key a unique message ID then press ENTER.

WB09012. KEY NEW MESSAGE ID AND PRESS "ENTER" TO COPY MESSAGE.

While copying a message definition, this message appears to prompt for the message ID of the new message definition.

ACTION: Key the new message definition ID then press ENTER. The new message definition will then be added.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DEMONSTRATION PROGRAM MESSAGES, “WD” Prefix

Messages produced by WNDODEMO (the demonstration feature) are prefixed by “WD” followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WD03001. PROGRAM xxxxxxxx NOT FOUND

The program designated by xxxxxxxx in the message is not installed.

ACTION: review the installation. Either the program is disabled, it was not link-edited or the program definition to CICS is not present.

ED03002. TRANSACTION CODE xxxx NOT DEFINED

The transaction designated by xxxx in the message is not installed.

ACTION: Review the installation. Either the transaction is disabled or the transaction definition to CICS is not present.

WD03003. FILE xxxxxxxx NOT DEFINED

The file designated by xxxxxxxx in the message is not installed.

ACTION: Review the installation. Either the file is disabled or the file definition to CICS is not present.

ED03005. x.x RELEASE OF PROGRAM pppppppp INSTALLED. MUST BE Y.Y

The program designated by pppppppp in the message is the wrong version.

ACTION: Review the installation. The designated program is for a prior release of CICS-WINDOWS, which means the replacement program of this release was not installed, or CICS is pointing to a load library containing some old program modules. In the message, x.x. indicates the release level of the program that was found, y.y indicates the release level it should be at.

ED03006. x.x CICS RELEASE OF PROGRAM pppppppp INSTALLED. INVALID FOR THIS CICS

The program designated by pppppppp in the message is the wrong CICS version.

ACTION: Review the installation. The designated program is the module intended for the release of CICS indicated by x.x in the message. It is incompatible with the release of CICS where the demonstration program is being executed.

WD03007. PROGRAM xxxxxxxx IS NOT DEFINED AS ASSEMBLER

The program designated by xxxxxxxx in the message is incorrectly defined.

ACTION: Review the installation. The definition of this program to CICS does not specify ASSEMBLER as the program language.

WD03008. TRANCODE xxxx POINT TO pppppppp INSTEAD OF yyyyyyyy

The transaction code indicated by xxxx in the message is incorrectly defined.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ACTION: Review the installation. The definition for this transaction should point to the program indicated by yyyyyyyy in the message. It currently points to pppppppp.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WD03009. TRANSACTION xxxx MUST HAVE TWASIZE OF 32

The transaction code indicated by xxxx in the message is incorrectly defined.

ACTION: Review the installation. The definition for this transaction must have a Transaction Work Area (TWASIZE) of at least 32 bytes.

WD03010. FILE xxxxxxxx HAS STRNO LESS THAN 3

The file indicated by xxxxxxxx in the message is incorrectly defined.

ACTION: Review the installation. The definition for this file must have at least three strings (STRNO).

WD03011. FILE xxxxxxxx MUST BE RECFORM VARIABLE

The file indicated by xxxxxxxx in the message is incorrectly defined.

ACTION: Review the installation. The definition for this file must have variable record format (RECFORM VARIABLE).

WD03012. CICS TCT SPECIFIES NO EXTENDED ATTRIBUTES. DEMO WILL BE INEFFECTIVE

The terminal where the demonstration is being executed does not support extended attributes.

ACTION: While you can run the demo on this terminal, it will not be as effective, since much of the demonstration uses extended color and highlighting. You should move to a terminal which supports extended attributes.

WD03013. CICS TCT SPECIFIES NO EXTENDED COLOR. THE DEMO WILL BE INEFFECTIVE

The terminal where the demonstration is being executed does not support extended color.

ACTION: While you can run the demo on this terminal, it will not be as effective, since much of the demonstration uses extended color. You should move to a terminal which supports extended color.

WD03014. CICS TCT SPECIFIES NO EXTENDED HIGHLIGHTING. DEMO WILL BE INEFFECTIVE

The terminal where the demonstration is being executed does not support extended highlighting.

ACTION: While you can run the demo on this terminal, it will not be as effective, since much of the demonstration uses extended highlighting. You should move to a terminal which supports extended highlighting.

WD03015. NO PROGRAM SYMBOL SUPPORT IN TCT. GRAPHIC ESCAPE WINDOWS SUPPRESSED

The terminal where the demonstration is being executed does not support program symbol tables.

ACTION: The demonstration makes use of the graphic escape character to draw its explanation boxes. Without program symbol support, it will use dashes and vertical bars.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WD04001. PROGRAM MODULE “xxxxxxxx” NOT FOUND

The program designated by xxxxxxxx in the message is not installed.

ACTION: Review the installation. Either the program is disabled, it was not link-edited or the program definition to CICS is not present.

WD04002. UNABLE TO START CICS-WINDOWS. TRY “WNDO.ON” ONLY

An internal attempt to activate CICS-WINDOWS has failed. This is probably an installation problem.

ACTION: Clear the screen and enter WNDO,ON (do not key DEMO). If you receive the User Configuration Display with no errors, WNDO,OFF and try the demo again. If errors occur, take action to correct the situation, then retry the demo.

WD04003. PROFILE “DEMO” HAS BEEN ALTERED. MUST HAVE 4 SESSIONS, 4 WINDOWS

A User Profile was found with the name “DEMO”, which is not defined correctly for the demonstration program.

ACTION: Delete or rename this profile. The demonstration program will dynamically build the correct profile if there is not one present.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDOHELP MESSAGES "WH" PREFIX

Messages produced by WNDOHELP (the HELP-WINDOWS feature) are prefixed by "WH" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WH01001. NO HELP TEXT IS AVAILABLE. PRESS CLEAR TO RESTORE TRANSACTION.

No help text that matched the current display was found.

ACTION: Press CLEAR to restore the current transaction.

WH01002. UNEXPECTED NO RECORD FOUND - xxxxxxxxxxxxxxxxxx

An unexpected no-record-found condition has occurred. The x's in the message are the text record key in question.

ACTION: This is probably an internal program error. Notify Unicom Systems technical support department.

WH01004. TERMINAL HAS BEEN SET TO xxxxxxxxxx TRANSLATE MODE

This message indicates successful completion of the UC command to reset the upper or lower case translation. The xxxxxxxxx in the message will be UPPER CASE or LOWER CASE.

ACTION: No action is necessary. You may proceed with any on-line activity.

WH01005. "xxxxxxxxx" IS INVALID IN THIS CONTEXT

The xxxxxxxxx in the message is an extraneous or undefined parameter for the command being issued.

ACTION: Correct the command and re-enter.

WH01007. DEFINE MODE IS xxx

This message is issued in response to the WHLP,DEFINE command. The xxx in the message is either ON or OFF, and specifies whether this terminal is eligible for defining help.

ACTION: No action is required.

WH01008. WINDOWS MUST BE ACTIVE ON THIS TERMINAL FOR DEFINE MODE

An attempt to activate define mode for this terminal has failed because CICS-WINDOWS is not active on this terminal.

ACTION: You must first activate CICS-WINDOWS on this terminal. See the WNDO,ON command.

WH01009. CUSTOMER IS NOT LICENSED TO USE THE HELP-WINDOW FEATURE

The HELP-WINDOWS feature is an additional paid feature of CICS-WINDOWS.

ACTION: If you are licensed to use the HELP-WINDOWS feature, contact Unicom Systems Technical support. If you wish to be licensed, contact your sales representative.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH02001. INVALID COMMAND CODE - xxxxxxxx

The command indicated by xxxxxxx is not a recognized command or keyword.

ACTION: Correct the command and re-enter.

WH02002. KEYWORD xxxxxxxx REQUIRES =DATA

The keyword indicated by xxxxxxx of the command is syntactically incorrect. This keyword must be followed by and equals sign (=) and an operand.

ACTION: Correct the keyword and re-enter.

WH02003. xxxxxxxx IS AN INVALID PF KEY MNEMONIC

The keyword indicated by xxxxxxx is preceded by 'PF' or 'PA' but is not a valid PF key mnemonic code, or it specifies a PF or PA key which is not valid in HELP-WINDOWS.

ACTION: Correct the keyword and re-enter.

WH02004. KEYWORD xxxxxxxx CANNOT BE FOLLOWED BY =DATA

The keyword indicated by xxxxxxx of the command is syntactically incorrect. This keyword must not be followed by and equals sign (=) and an operand.

ACTION: Correct the keyword and re-enter.

WH02005. NO DATA FOLLOWING = FOR KEYWORD xxxxxxx

The keyword indicated by xxxxxxx of the command is syntactically incorrect. There is no data following the equal sign (=) or the first character after the equal sign is a space.

ACTION: Correct the keyword and re-enter.

WH02006. DATA (xxxxxxx) FOR KEYWORD yyyyyyy MUST BE NUMERIC

The data following the equal sign indicated by xxxxxxxx in the message for the keyword indicated by yyyyyyy is syntactically incorrect. The data must be all numeric.

ACTION: Correct the keyword and re-enter.

WH02007. DATA (xxxxxxx) FOR KEYWORD yyyyyyy IS TOO LONG

The data following the equal sign indicated by xxxxxxxx in the message for the keyword indicated by yyyyyyy is syntactically incorrect. The number of characters in the data field exceeds the maximum allowable for this keyword.

ACTION: Correct the keyword and re-enter.

WH11001. CROSS-DOCUMENT FIELD IS ALREADY ON FILE

The cross-document record that you are attempting to add already exists in the file.

ACTION: Change the system or field name and re-enter.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH11002. CROSS-DOCUMENT FIELD IS NOT ON FILE

The cross-document record that you are attempting to retrieve does not exist in the file.

ACTION: Correct the record name and re-enter. If you think the record is there, try the NEXT command, then browse forward or backward until you find it, or use the directory to locate it.

WH11003. PRESS "ENTER" TO UPDATE OR PRESS THE "QUIT" KEY

This message confirms entry into update mode for a cross-document record. You can now make changes to the record.

ACTION: Make any desired changes and press ENTER to complete the update, or press the QUIT key to exit update mode.

WH11004. PRESS "ENTER" TO DELETE OR PRESS THE "QUIT" KEY

This message confirms entry into delete mode for a cross-document record. The requested record is displayed on the screen.

ACTION: Press ENTER to complete the delete, or press the QUIT key to exit deletion mode.

WH11005. ENTER DATA AND PRESS "ENTER" TO ADD OR PRESS THE "QUIT" KEY

This message confirms entry into add mode for a cross-document record. You can now enter all data fields for the record.

ACTION: Enter all required fields and press ENTER to complete the add, or press the QUIT key to exit add mode.

WH11006. CROSS-DOCUMENT HAS BEEN ADDED

This message confirms successful completion of a record add for a cross-document record.

ACTION: No action is required, continue with the next maintenance function.

WH11007. CROSS-DOCUMENT RECORD HAS BEEN CHANGED

This message confirms successful completion of a record update for a cross-document record.

ACTION: No action is required, continue with the next maintenance function.

WH11008. CROSS-DOCUMENT RECORD HAS BEEN DELETED

This message confirms successful completion of a record delete for a cross-document record.

ACTION: No action is required, continue with the next maintenance function.

WH11009. END OF X-DOCS USE "PREV" OR "QUIT" KEY

While browsing forward through cross-document records, the end of these record has been reached.

ACTION: You can browse backward with a 'P' command, request help with the help key, or press the QUIT key to exit.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH11010. ENTER KEY FIELDS TO RETRIEVE RECORD FOR DELETION

A DELETE command has been issued for a cross-document record and the system and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record to be deleted.

WH11011. SYSTEM HAS NOT BEEN DEFINED

The cross-document record that you are attempting to add contains a system identifier for which no system record exists. You can not add a lower-level record without a complete hierarchy.

ACTION: Correct the system name and re-enter, or add the necessary system record.

WH11012. DOCUMENT HAS NOT BEEN DEFINED

The DOCUMENT field of the cross-document record names a document identifier for which no document record exists.

ACTION: Correct the document name and re-enter, or add the necessary document record.

WH11013. SECTION HAS NOT BEEN DEFINED

The SECTION field of the cross-document record names a section identifier for which no section record exists.

ACTION: Correct the section number and re-enter, or add the necessary section record.

WH11014. SUBJECT HAS NOT BEEN DEFINED

The SUBJECT field of the cross-document record names a subject identifier for which no subject record exists.

ACTION: Correct the subject name and re-enter, or add the necessary subject record.

WH11015. NO RECORD FOR THIS KEY. ENTER VALID KEY AND PRESS "ENTER"

An INQUIRE or UPDATE command has been issued for a cross-document record and the system and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record.

WH11016. KEY FIELDS HAVE BEEN CHANGED, RECORD HAS NOT BEEN UPDATED

An UPDATE command was issued, but prior to pressing ENTER, one or more of the identifiers in the command was changed.

This message indicates that the record update did not take place.

ACTION: No action is necessary, continue with any activity.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH12001. THIS RECORD IS ALREADY ON FILE. CORRECT KEY OR PRESS THE "QUIT" KEY

The transaction key record that you are attempting to add already exists in the file.

ACTION: Change the system or field name and re-enter, or press the QUIT key to exit add mode.

WH12002. TRAN KEY IS NOT ON FILE

The transaction key record that you are attempting to retrieve does not exist in the file.

ACTION: Correct the record name and re-enter. If you think the record is there, try the NEXT command, then browse forward or backward until you find it, or use the directory to locate it.

WH12003. NO RECORD FOR THIS KEY, ENTER VALID KEY AND PRESS "ENTER"

An INQUIRE or UPDATE command has been issued for a transaction key record and the transaction ID, screen and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record.

WH12004. PRESS "ENTER" TO UPDATE

This message confirms entry into update mode for a transaction key record. You can now make changes to the record.

ACTION: Make any desired changes and press ENTER to complete the update, or press the QUIT key to exit update mode.

WH12005. PRESS "ENTER" TO DELETE OR THE "EXIT" KEY

This message confirms entry into delete mode for a transaction key record. The requested record is displayed on the screen.

ACTION: Press ENTER to complete the delete, or press the QUIT key to exit deletion mode.

WH12006. ENTER DATA AND PRESS "ENTER" TO ADD OR THE "EXIT" KEY

This message confirms entry into add mode for a transaction key record. You can now enter all data fields for the record.

ACTION: Enter all required fields and press ENTER to complete the add, or press the QUIT key to exit add mode.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH12007. TRAN KEY HAS BEEN ADDED

This message confirms successful completion of a record add for a transaction key record.

ACTION: No action is required, continue with the next maintenance function.

WH12008. TRAN KEY HAS BEEN CHANGED

This message confirms successful completion of a record update for a transaction key record.

ACTION: No action is required, continue with the next maintenance function.

WH12009. TRAN KEY HAS BEEN DELETED

This message confirms successful completion of a record delete for a transaction key record. Note that pressing ENTER after a deletion will result in a no-record-found condition. Browsing backward will produce the same result. The only file activity that can be done without changing the command line is to browse forward.

ACTION: You may browse forward with the NEXT key, request help with the help key, or press the QUIT key to exit.

WH12010. END OF TRAN KEYS. USE "PREV" OR "EXIT" KEY

This message indicates that the end of transaction key records has been reached while browsing forward with the NEXT command.

ACTION: You may browse backward with the PREV key, request help with the help key, or press the QUIT key to exit.

WH12011. ENTER KEY FIELDS TO RETRIEVE RECORD FOR DELETION

A DELETE command has been issued for a transaction key record and the transaction ID, screen and field names were not entered, nor were they present in the conversation.

This message can also occur if a name is filled-in from the conversation and the resultant record key does not exist.

ACTION: Enter the necessary key fields to retrieve the desired record to be deleted.

WH12012. SYSTEM HAS NOT BEEN DEFINED

The SYSTEM field for the cross-document identity in the transaction key record contains a system identifier for which no system record exists.

ACTION: Correct the system name and re-enter, or add the necessary system record.

WH12013. X-DOC FIELD IS INVALID. PRESS "ENTER" TO CONFIRM OR "EXIT"

The cross-document identity in the transaction key record does not exist.

ACTION: Correct the cross-document name or system and re-enter, or go ahead and press enter to add this transaction key record with an invalid cross-document identifier.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH12014. THE "FROM" RECORD COULD NOT BE FOUND. PRESS "EXIT" TO END COPY

The starting record of a COPY function does not exist as entered.

ACTION: Restart the COPY command and enter a valid starting record identifier or press the QUIT key to exit the copy function.

WH12015. TRAN KEY TABLE EXHAUSTED. INCREASE THE TKEYPARM IN HWINTBL

This message should not occur.

ACTION: First try backing-out CICS-WINDOWS, then reinitialize WINDOWS and retry the operation that failed. If the message still occurs, contact Unicom Systems Technical Support.

WH13001. CROSS-DOCUMENT "xxxxxxxxxx" IS NOT DEFINED

The cross-document identity in the transaction key record or as the object of an INCLUDE statement does not exist. The xxxxxxxx in the message identifies the cross-document name.

ACTION: Either add the missing cross-document record or change the reference in the transaction key or the INCLUDE statement.

WH13002. TRANSACTION KEY "xxxxxxxxxx" IS NOT DEFINED

The transaction key denoted by xxxxxxxxxx does not exist.

ACTION: Retry the operation using the Transaction Key directory.

WH13003. THE LAST PAGE OF THIS SECTION/SUBJECT HAS BEEN DISPLAYED

This message indicates that the end of text records for this section or subject has been reached while browsing forward with the NEXT command.

ACTION: No action is required, continue with the next maintenance function.

WH13004. THE FIRST PAGE OF THIS SECTION/ SUBJECT HAS BEEN DISPLAYED

This message indicates that the beginning of records for this section or subject has been reached while browsing backward with the PREVIOUS command.

ACTION: No action is required, continue with the next maintenance function.

WH13005. TEXT RECORD IS NOT ON FILE

The text record requested with a LIST or GET command is not on file.

ACTION: Correct the record number and re-enter, or use the NEXT and PREVIOUS commands to browse the file.

WH13006. <INCLUDE> STATEMENT IS RECURSIVE AND HAS BEEN IGNORED

An INCLUDE statement has been found which references a cross-document record which points to the text records currently being processed.

ACTION: No action is necessary. You may want to locate this INCLUDE statement an correct the command.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH13007. <INCLUDE> STATEMENT EXCEEDS MAXIMUM ALLOWABLE INCLUDES

The level of nested INCLUDE statements has exceeded the maximum allowed.

ACTION: No action is necessary, however, the text referenced by this INCLUDE statement will not be processed.

WH13008. <INCLUDE> STATEMENT REFERENCES AN INVALID TEXT RECORD

An INCLUDE statement has been found which references a cross-document record which points to a text record that is not on file.

ACTION: No action is necessary. You may want to locate this INCLUDE statement and correct the text reference fields of the cross-document record.

WH13009. <INCLUDE> STATEMENT REFERENCES AN UNDEFINED CROSS-DOCUMENT

An INCLUDE statement has been found which references a cross-document record which is not on file.

ACTION: No action is necessary. You may want to locate this INCLUDE statement an correct the cross-document name, or add the referenced cross-document record.

WH13010. <INCLUDE> STATEMENT IS SPECIFIED INCORRECTLY

An INCLUDE statement has been found which is syntactically incorrect.

ACTION: No action is necessary. You may want to locate this INCLUDE statement an correct it.

WH13011. ALL TEXT FOR FIELD "xxxxxxxxx" HAS BEEN DISPLAYED

This message indicates that the end of text records for this cross-document record has been reached while using the GET command. The xxxxxxx in the message is the cross-document name.

ACTION: No action is required, continue with the next maintenance function.

WH13012. USER EXIT "xxxxxxxx" IN <USERXIT> COMMAND CANNOT BE FOUND

A USERXIT command has been found while processing text records in help access mode. The designated program could not be loaded into memory. The xxxxxxx in the message is the program name.

ACTION: Verify that that the exit program is present in one of the load libraries available for this CICS system. Also verify that a PPT entry is present with the same program name. If problem persists, notify Unicom Systems technical support.

WH13013. RETURNED PARAMETERS FROM USER EXIT "xxxxxxxx" ARE INVALID, RCODE=y

Upon return from a user exit program specified with a USERXIT command in the text, the return parameters set by the exit program are incorrect. The xxxxxxx in the message is the program name. The 'y' in the message is a code indicating the type of error. Meanings of the code are:

1) Invalid status code set.
2) The returned cross-document record is invalid.
3) The returned text key is invalid.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ACTION: Review the specifications for writing a user exit program in the TECHNICAL REFERENCE manual. Correct the errors in the program and re-initialize it. If you need help determining the error, notify Unicom Systems technical support.

WH14001. TEXT RECORD NUMBER IS REQUIRED

A text record addition was attempted and no text record number was entered, nor is one present in the conversation.

ACTION: Enter the REC= keyword to supply the text record number and re-enter the command.

WH14002. DOCUMENT MUST BE REORGANIZED BEFORE THIS RECORD CAN BE INSERTED

During text record addition, the next text record after the one previously added contains a record number which is only one greater than the previous number. You cannot add any more records at this point without re-numbering the section.

ACTION: If more text records must be added at this point, use the RENUMBER command to reorganize the text records for this section/subject.

WH14003. PRESS ENTER TO ADD RECORD OR THE "QUIT" KEY

This message confirms entry into add mode for a text record. You can now enter all data fields for the record.

ACTION: Enter all required text data and press ENTER to complete the add, or press the QUIT key to exit add mode.

WH14004. TEXT RECORD NUMBER ALREADY EXISTS

The text record assigned to the record that you are attempting to add already exists in the file.

ACTION: Change the record number, or press PF3 to exit add mode.

WH14005. MAKE CHANGES AND PRESS "ENTER" TO UPDATE RECORD

This message confirms entry into edit mode for a text record. You can now make changes to the record, if desired.

ACTION: Make any desired changes and press ENTER to complete the update, or perform any other function.

WH14006. REQUESTED RECORD IS NOT ON FILE

The text record that you are attempting to retrieve does not exist in the file.

ACTION: Correct the record name and re-enter. If you think the record is there, try the NEXT command, then browse forward or backward until you find it, or use the directory to locate it.

WH14007. PRESS "ENTER" TO DELETE RECORD OR THE "QUIT" KEY

This message confirms entry into delete mode for a text record. The requested record is displayed on the screen.

ACTION: Press ENTER to complete the delete, or press PF3 to exit deletion mode.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH14012. TEXT RECORD HAS BEEN ADDED

This message confirms successful completion of a record add for a text record.

ACTION: No action is required, continue with the next maintenance function.

WH14013. TEXT RECORD HAS BEEN CHANGED

This message confirms successful completion of a record update for a text record.

ACTION: No action is required, continue with the next maintenance function.

WH14014. TEXT RECORD HAS BEEN DELETED

This message confirms successful completion of a record delete for a text record.

ACTION: No action is required, continue with the next maintenance function.

WH14015. THE LAST TEXT RECORD IN THIS SECTION/SUBJECT HAS BEEN DISPLAYED

This message indicates that the end of text records for this section or subject has been reached while browsing forward with the NEXT command.

ACTION: No action is required, continue with the next maintenance function.

WH14016. THE FIRST TEXT RECORD IN THIS SECTION/SUBJECT HAS BEEN DISPLAYED

This message indicates that the beginning of text records for this section or subject has been reached while browsing backward with the PREVIOUS command.

ACTION: No action is required, continue with the next maintenance function.

WH14023. PRESS ENTER TO RENU_MBER TEXT RECORDS OR THE "QUIT" KEY

This message confirms entry into renumber mode for a series of text records.

ACTION: Press ENTER to start the renumber process, or press the QUIT key to exit renumber mode.

WH14024. TEXT RECORDS HAVE BEEN RENUMBERED

This message indicates successful completion of the RENUMBER command.

ACTION: No action is necessary, continue with any function.

WH14025. TEXT HAS BEEN FOUND, PRESS "ENTER" TO CONTINUE SEARCH

This message indicates that the data entered as the operand of a SEARCH command has been located. The text record containing the search data is displayed.

ACTION: Press ENTER to continue searching forward in the text from this point, otherwise continue with any function.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH14026. ALL TEXT HAS BEEN SEARCHED

This message indicates that the end of text records for this section/subject was encountered before the data entered as the operand of a SEARCH command could be found. The search data does not exist in this text in the form that it was entered.

If the search data was previously found, it means that there are no more occurrences of the search data after the last one that was found.

ACTION: Reposition to the beginning of the text for this section/subject and restart the search with a different search string, if desired, otherwise continue with any function.

WH14027. TEXT RECORD REQUIRES ALTERNATE SCREEN SIZE CAPABILITY

The text record being reviewed was created on a terminal with a larger screen size than the terminal attempting to display it and the ALTSCRN option was set when the text record was created. Or, the TCT entry for this terminal does not specify the same alternate screen size as the terminal which created this text record.

ACTION: You cannot view this text record as it is from this terminal. Either use a terminal with the correct alternate screen size or update the text record, removing the ALTSCRN option.

WH14028. xxxxxxxx HEADER RECORD MISSING, CANNOT ADD TEXT

A required hierarchical control record is not present for the system, document, section, subject hierarchy as specified. Text addition cannot be done until the complete hierarchy is present. The xxxxxxxx in the message indicates the type of control record that is missing.

ACTION: Add the required control record, then restart the text record addition function.

WH14029. TRANSACTION KEY "xxxxxxxxxxxxxxxxx" IS NOT VALID

The transaction key denoted by xxxxxxxxxx does not exist.

ACTION: Retry the operation using the Transaction Key directory.

WH18002. THE LAST RECORD HAS BEEN DISPLAYED

This message indicates that all records have been displayed for this directory display.

ACTION: No action is necessary, continue with any function.

WH18003. TRAN KEY "xxxxxxxxxxxxxxxxx" WAS NOT FOUND

While attempting to retrieve text from the transaction key directory, the designated transaction key record could not be found.

ACTION: This is probably a system error although it could be caused by incorrect input on the command line. Attempt the operation again and if the problem persists, notify Unicom Systems technical support..

WH24001. CROSS-DOCUMENT FIELD IS TOO LONG

The cross-document name exceeds the maximum length allowed. The maximum length is 20 characters

ACTION: Correct the field name command and re-enter.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH24002. AN ERROR HAS BEEN ENCOUNTERED IN THE TEXT COMMAND AREA

The edit sub-command in the text command area is not a valid sub-command.

ACTION: Correct the sub-command and re-enter.

WH24003. <SPACE NN> COMMAND IS SPECIFIED INCORRECTLY

A text command identified as SPACE was found but cannot be processed as it is syntactically incorrect.

ACTION: Correct the SPACE command and re-enter.

WH24004. NO STARTING POSITION HAS BEEN DEFINED FOR CROSS-DOCUMENT

A STOPFLD statement has been encountered naming a cross-document for which no STRTFLD was present.

ACTION: Enter the STRTFLD statement or change the name of the field in the STOPFLD statement if it is incorrect.

WH24005. <STRTFLD> HAS ALREADY BEEN DEFINED FOR THIS CROSS-DOCUMENT

A STRTFLD statement has been encountered naming a cross-document which already has a STRTFLD in the file.

ACTION: Change the name of the field in the STRTFLD statement to a different cross-document name, or delete the duplicate STRTFLD statement.

WH24006. <STOPFLD> HAS ALREADY BEEN DEFINED FOR THIS CROSS-DOCUMENT

A STOPFLD statement has been encountered naming a cross-document which already has a STOPFLD in the file.

ACTION: Change the name of the field in the STOPFLD statement to a different cross-document name, or delete the duplicate STOPFLD statement.

WH24007. SYSTEM ID IS INVALID FOR <STRTFLD>

A STRTFLD statement has been encountered which names a system ID as well as a cross-document field name. The named system ID does not exist.

ACTION: Change the name of the system in the STRTFLD statement to a different name, or add a valid system record with the requested name.

WH24008. SYSTEM ID IS INVALID FOR <STOPFLD>

A STOPFLD statement has been encountered which names a system ID as well as a cross-document field name. The named system ID does not exist.

ACTION: Change the name of the system in the STOPFLD statement to a different name, or add a valid system record with the requested name.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH24009. SPLIT COMMAND IS NOT VALID FOR TRANSACTION KEY TEXT

You cannot use the SPLIT command when adding or editing fast-path help text. Fast-path text consists of one text record which is attached to the transaction key record. If you need additional text, either change to structured text or add another transaction key record with a higher sequence number.

ACTION: Remove the SPLIT command and take appropriate action as described above.

WH24010. <STRTFLD> IS INVALID FOR TRANSACTION KEY TEXT

You cannot use the STRTFLD or STOPFLD command when adding or editing fast-path help text. Fast-path text consists of one text record which is attached to the transaction key record. Therefore it is not necessary to define a cross-document record.

ACTION: Remove the STRTFLD command and continue with text maintenance.

WH24011. INSERT WOULD EXCEED TWO PAGES, COMMAND IGNORED

An INSERT sub-command as been used when editing fast-path help text. There is insufficient room in the text record to add insert all of the data that was copied or moved. Fast-path text consists of one text record which is attached to the transaction key record. The maximum size is the size of one VSAM record. If you need additional text, either change to structured text or add another transaction key record with a higher sequence number.

ACTION: Remove the INSERT command and take appropriate action as described above.

WH24012. CROSS-DOCUMENT FIELD HAS NOT BEEN DEFINED

The cross-document name specified on an INCLUDE statement could not be located as coded.

ACTION: Verify that the cross-document record exists within the specified system. Correct the INCLUDE statement or add the cross-document record.

WH24013. EXIT NAME MUST BE SPECIFIED IN <USERXIT> COMMAND

A USERXIT statement was found in the text which did not specify a program name.

ACTION: Enter the program name following the USERXIT command. Enclose the entire statement in <> signs.

WH24014. EXIT PARAMETERS IN <USERXIT> COMMAND ARE TOO LONG

The PARM statement following the program name in a USERXIT command exceeds the allowable maximum.

ACTION: Correct the PARM statement. Review the specifications for the USERXIT command if needed.

WH24015. ENDING BLOCK COMMAND NOT ENTERED FOR "xx"

A block deletion, move or copy function was specified by keying a DD, MM or CC in the sub-command area of the editor. The ending DD, MM or CC was omitted when ENTER was pressed.

ACTION: Key the ending block command or remove the starting block command. Note that you cannot perform block functions that span more than one text record.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WH24016. FIELD NAME IS REQUIRED IN STRTFLD COMMAND

A STRTFLD statement was found in the text which did not specify a name.

ACTION: Enter the name following the STRTFLD command. Enclose the entire statement in <> signs.

WH24017. FIELD NAME CANNOT CONTAIN "," OR "="

The field name cannot contain the indicated characters.

ACTION: Correct the field name and press ENTER.

WH25001. TEXT RECORD REORGANIZATION IS IN PROGRESS. PASS x

This is an informational message to report the status of a text record reorganization.

ACTION: No action is required.

WH25002. THERE ARE NO RECORDS IN THIS SECTION/SUBJECT TO RENUMBER

While attempting to perform a text record reorganization, the function was halted because there are no records to reorganize.

ACTION: No action is required

WH25003. RENUMBER COULD NOT BE COMPLETED BECAUSE X-DOC RECORD WAS MISSING

While attempting to perform a text record reorganization, the function was halted because a cross document record is missing.

ACTION: Manually add the cross document record.

WH25004. RENUMBER COULD NOT BE COMPLETED BECAUSE TEXT RECORD WAS MISSING

While attempting to perform a text record reorganization, the function was halted because a cross document record pointed to a text record that no longer exists.

ACTION: Manually delete or correct the bogus cross document record.

WH25005. RENUMBER COULD NOT BE COMPLETED BECAUSE TEXT RECORD WAS DUPLICATED

During the processing of a RENUMBER command, a text record could not be added because the generated record number already exists. The records are partially renumbered at this point.

ACTION: This is probably a system error. The situation should not occur because unique record numbers are calculated during the renumber operation. Notify Unicom Systems technical support.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDOINIT MESSAGES "WI" PREFIX

Messages produced by WNDOINIT (the WNIT transaction) are prefixed by "WI" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WI01005. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-WINDOWS could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WI02001. THIS TERMINAL IS NOT USING CICS-WINDOWS.

While attempting to display the CICS-WINDOWS profile of the user of this terminal, the function was not completed because this terminal is not using CICS-WINDOWS.

ACTION: Press PF3 to exit.

WI02002. THIS PROFILE IS PROTECTED. WINDOW CONFIGURATION CANNOT BE SAVED.

From the User Configuration Display, an attempt to save changes made to the configuration has failed because the corresponding Profile is protected.

ACTION: The changes cannot be saved from this screen. If the changes are to be permanent, the User Profile must be altered.

WI02003. PROFILE HAS BEEN DELETED FROM OPTION TABLE. IT CANNOT BE SAVED.

From the User Configuration Display, an attempt to save changes made to the configuration has failed because the corresponding Profile no longer exists.

ACTION: The profile must be re added in order to save the changes.

WI02004. WINDOW CONFIGURATION HAS BEEN SAVED.

This message indicates that the attempt to save changes made to the configuration has successfully completed.

ACTION: No action is required.

WI04002. "xxxxxxxx" IS NOT A VALID WINDOWS COMMAND.

While editing the CICS-WINDOWS PF key assignments, an invalid CICS-WINDOWS command was attempted to be assigned to a PF key.

ACTION: Review the valid CICS-WINDOWS commands, and correct the error.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MENU GENERATION MESSAGES "WM" PREFIX

Messages produced by WNDOMENU are prefixed by "WM" followed by a 5-position message number, then the message text. The full message will be listed, followed by an explanation and appropriate action to take.

WM02001. MENU "xxxxxxxx" HAS NOT BEEN DEFINED.

The attempted function failed because the menu definition could not be located in the file being searched.

ACTION: If multiple menu files are in use, the menu definition may reside in another file. In which case, you should ensure that the file is defined to CICS-WINDOWS. For more information see section 11 - *CUSTOMIZATION*, under *THE FILE TABLE*. If multiple menu files are not in use, then the menu definition does not exist.

WM03002. PROGRAM "xxxxxxxx" IS NOT AVAILABLE.

CICS-WINDOWS could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WM03003. TRANSACTION "xxxx" IS NOT AVAILABLE.

CICS-WINDOWS could not find the transaction xxxx. Either the PCT entry is disabled or not present.

ACTION: Ensure that the PCT entry is available to CICS-WINDOWS.

WM03004. WINDOWS COMMAND "xxxxxxxx" COULD NOT BE EXECUTED.

The selected menu option has a 'Window Command' associated with it. The command was not executed because the command is syntactically incorrect or an invalid function at this point in windowing operation.

ACTION: Correct the Window Command in the Menu Definition.

WM04004. THE FIRST MENU DEFINITION HAS BEEN DISPLAYED.

While browsing backward through the menu definitions, the first definition on file has been displayed.

ACTION: To browse through more entries you must browse in a forward direction.

WM04005. THE LAST MENU DEFINITION HAS BEEN DISPLAYED.

While browsing forward through the menu definitions, the last definition on file has been displayed.

ACTION: To browse through more entries you must browse in a backward direction.

WM05006. MENU "xxxxxxxx" HAS BEEN DELETED.

This message merely confirms that menu definition xxxxxxxx was successfully deleted from the file.

ACTION: No action is required.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WM05007. KEY DATA AND PRESS "ENTER" TO ADD MENU.

In response to New function (creating a new menu definition), this message is prompting you to key the desired data then press ENTER. The menu will then be added.

ACTION: Key desired data then press ENTER.

WM05008. MENU "xxxxxxxx" HAS BEEN ADDED.

This message merely confirms that the menu definition xxxxxxxx was successfully added to the file.

ACTION: No action is required.

WM05009. MENU "xxxxxxxx" HAS BEEN CHANGED.

This message merely confirms that the menu definition xxxxxxxx was successfully altered.

ACTION: No action is required.

WM05010. MENU "xxxxxxxx" WAS NOT FOUND.

In response to a Find function, CICS-WINDOWS was unable to locate the menu definition xxxxxxxx in the file being searched.

ACTION: If multiple files are being used for CICS-WINDOWS then the definition may exist in another file.

WM05011. MENU ID "xxxxxxxx" ALREADY EXISTS.

A request to add a new menu definition failed because an old definition with the same menu ID already exists.

ACTION: Key a unique menu ID then press ENTER.

WM05012. KEY NEW MENU ID AND PRESS "ENTER" TO COPY MENU.

While copying a menu definition, this message appears to prompt for the menu ID of the new definition.

ACTION: Key the new menu definition ID then press ENTER. The new menu definition will then be added.

WM05013. NO MENU SELECTION AT CURSOR. PRESS "ENTER" TO ADD.

While in the Menu Editor, the SELECT key was pressed. There was not a cursor selectable menu selection at the position of the cursor when the SELECT key was pressed.

ACTION: If you wish to create a cursor selectable menu selection at the position of the cursor, key all desired fields in the select window, then press ENTER.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDOMRO MESSAGES "WMR" PREFIX

Messages produced by the WNDOMRO program, which runs in a remote region in an MRO/ISC environment are prefixed by "WMR".

WMR0100. WNDOMRO SUCCESSFULLY ACTIVATED

This message indicates that WNDOMRO is successfully installed and ready for operation.

ACTION: No action is required, you can begin operation of CICS-WINDOWS now.

WMR0200. NO PSEUDO IDS FOUND IN AUTOTBL, WNDOMRO IS NOT NEEDED

This message appears on the system console during CICS startup for a remote region CICS. It indicates that one of the following conditions exists:

1) No User Option file (WNDOFIL) was found in the Core-image/Load library.

2) The User Option Table does not contain any Auto-Init table statements.3). All Auto-Init table statements have PSEUDO IDS=NO code.

ACTION: If Pseudo Terminal IDs are not in use, there is no need to activate the WNDOMRO program in the remote region. If they are in use, you must have an Auto-Start table present in the remote region which defines the Pseudo IDs to be used. See *RUNNING CICS-WINDOWS IN AN MRO/ISC ENVIRONMENT* in section 13 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS*, for more information.

No action is needed at this point. CICS will be properly initialized and you may use CICS-WINDOWS, but Pseudo Terminal IDs will not work properly for any transactions in this remote region..

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

TEXT EDITOR MESSAGES "WN" PREx"

Messages produced by the Menu or Message text editor are prefixed by "WN" followed by a 5-position message number, then the message text.

WN19001. TEXT HAS EXCEEDED MAXIMUM RECORD LENGTH

The menu or message is to large.

ACTION: If this is a message and you have multiple consecutive blank lines, try using the SPACE command. For more information, see section 08, under *THE MENU TEXT EDITOR*.

WN19002. AN ERROR HAS BEEN ENCOUNTERED IN THE TEXT COMMAND AREA.

An invalid command has been issued.

ACTION: Review the appropriate command in section 08, under *THE MENU TEXT EDITOR*.

WN19001. INSERT WOULD EXCEED TWO PAGES. COMMAND IGNORED.

The attempted insert could not be completed because it was too large.

ACTION: Try ing/moving a smaller number of lines.

WN19001. ENDING BLOCK COMMAND NOT ENTERED FOR "xx".

A block command is in progress but has not been performed because the block command does not have a matching command.

ACTION: Review the appropriate command in section 08, under *THE MENU TEXT EDITOR*.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CICS-WINDOWS MESSAGES "WNDO" PREFIX

Messages produced by WINDOWS are prefixed by "WNDO" followed by a 4-position message number, then the message text.

For this document, the full message will be listed, followed by an explanation of appropriate action to take.

WNDO0001. ERROR RETURN FROM TEMP-STORAGE IO, CODE=xx.

A non-zero code was returned by CICS from a Temporary Storage data access. This problem can occur if a CICS task running under CICS-WINDOWS causes a storage violation, or if auxiliary Temporary Storage is in use and the control interval size of the Temporary Storage dataset is smaller than the largest terminal screen size plus 500 bytes.

ACTION: If problem persists, notify Unicom Systems.

WNDO0002. xxxx CAN'T BE STARTED WHILE yyyy IS ACTIVE IN SESSION z

This message can occur if the Single Occurring Transaction Table is installed and one of the transactions in the table is attempted to start in more than one virtual terminal. It can also occur if the transaction designated by yyyy in the message is defined as an 'excluded' transaction in the User Option Table (For more information, please refer to *SINGLE OCCURRING TRANSACTIONS TABLE, SINGLE OCCURRING PROGRAMS TABLE,* or *TRANSACTION EXCLUSION TABLE in section 11 - CUSTOMIZATION*).

In the message,

xxxx = The transaction code entered.
yyyy = The transaction or program name in conflict in the other session.
z = The virtual terminal number (session number) where the conflicting application is running.

ACTION: You can not start this transaction without first terminating the conflicting transaction in the other virtual terminal.

WNDO0003. ACTIVE TASK xxxx ON LOGICAL TERMINAL t MUST BE ENDED

This message occurs in response to a WNDO,OFF command. It states that you must terminate task xxxx in session t before the WNDO,OFF command may be performed. (This is a customization option, for more information see *FIELDS OF THE USER OPTIONS TABLE* in section 11 - *CUSTOMIZATION*).

ACTION: Toggle to session t and exit the task, then issue the WNDO,OFF command.

WNDO0004. DO YOU WANT TO TERMINATE CICS-WINDOWS ?

The operator has signed off via the CSSF (or some other) transaction. CICS-WINDOWS intercepts the sign-off transactions and gives this query. It is intended to serve as a reminder that CICS-WINDOWS is still active on this terminal.

ACTION: If you want to terminate CICS-WINDOWS on this terminal, press ENTER. (the "Y" response is pre-coded on the screen). Otherwise key an "N" and press ENTER.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO0005. CICS-WINDOWS PASSWORD ERROR, CODE=x

This message indicates that the product has expired or cannot be used because of a password error or installation error. The product is now inactive and cannot be used.

The "x" in the above message will be one of the following error codes:

1	-	The current date is greater than the password expiration date.
2	-	Module "STSPASS" could not be found in a library in the search string.
3	-	The product ID could not be located in STSPASS.
4	-	The password in STSPASS is invalid.
5	-	Module "STS0100" could not be found in a library in the search string.
6	-	STSPASS contains a CPU ID that is invalid for the machine that it is installed on.
7	-	The password entered is not valid for the requested product.
9	-	An undetermined error has been encountered. Please call Unicom Systems Technical Support at (818) 838-0606.

ACTION: Correct the error if possible, or call Unicom Systems Technical Support at (818) 838-0606.

WNDO0006. RELEASE LEVEL OF CICS IS INCOMPATIBLE WITH CICS-WINDOWS

This message appears if the wrong version of CICS-WINDOWS is installed; the CICS 1.6 version on a 1.7 CICS, for instance.

ACTION: Re-install CICS-WINDOWS.

WNDO0007. ffffffff ERROR ON aaaaaa, CODE=bb-cccccccc, VSAM CODE=dd

During a file control access to the VSAM file ffffffff, an error was returned from CICS. In the message,

ffffffff	=	The name of the VSAM file.
aaaaaa	=	The request macro being issued.
bb	=	The return code from CICS.
cccccccc	=	The interpreted CICS code.
dd	=	The VSAM error code if cccccccc is 'ILLOGIC'.

The following action is taken by CICS-WINDOWS on any error return during access to WNDOFIL:

1). The message is displayed on the terminal.
2). The message is displayed on the console.
3). A transaction dump is taken.

ACTION: For an explanation of the CICS return codes, refer to the *CICS APPLICATION PROGRAMMERS REFERENCE MANUAL* under *FILE CONTROL RESPONSE CODES*.

Attempt to correct the problem with the file and continue operation. Or, issue the WNDO,TEMP command to switch to using Temporary Storage.

WNDO0008. FILE 'xxxxxxxx' IS NOT OPEN

The VSAM file xxxxxxxx has been closed to CICS users.

ACTION: The file must be opened.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO0009. CICS-WINDOWS MUST BE TERMINATED TO LOG-OFF

You can not sign off of CICS without first terminating CICS-WINDOWS. This is an option specified in the User Option Table.

ACTION: Enter WNDO,OFF to terminate CICS-WINDOWS, then do the sign off.

WNDOO011. TRANSACTION xxxx NOT ALLOWED IN WINDOW MODE

You can not issue this transaction code while in window mode. You must be in full-screen mode. The message indicates that this transaction code is defined in the TYPE=WTXNTBL of the User Option Table (see *CUSTOMIZATION*).

This message will also appear if you enter a CSSF (or equivalent sign-off) transaction code while in window mode, since sign-off is not allowed from within a window.

ACTION: Press the 'WINDOW' key or use the "X" command to return to full-screen mode and then enter the transaction.

WNDO0012. WINDOWS CAN NOT BE INITIALIZED, JUGGLER HAS BEEN USED.

The CICS-WINDOWS product and the CICS-JUGGLER product can not run together on the same system. If CICS-JUGGLER is used before a WNDO transaction is entered, this message will result. If a WNDO transaction is performed first, the JUGL transaction is automatically disabled.

ACTION: You should remove the JUGL transaction from the CICS-PCT and bring CICS down and up.

WNDO0013. USER EXIT INTERFACE NOT PRESENT, NEED EXITS = YES IN DFHSIT.

In attempting to enable the After-Input TCP/ZCP user exit to CICS during CICS-WINDOWS initialization, it was determined that the CICS User Exit Interface is not enabled.

ACTION: You must specify EXITS=YES in the System Initialization Table or as a SIT override parameter in the CICS start-up JCL, then bring CICS down and up.

WNDO0014. ERROR HAS OCCURRED DURING EXIT ENABLE, EIBRCODE = X'nnnn'

In attempting to enable the After-Input TCP/ZCP user exit to CICS during CICS-WINDOWS initialization an error was returned by CICS from the ENABLE command. X'nnnn' is the 2-byte hexadecimal code posted in the EIBRCODE field by CICS. See the *CICS CUSTOMIZATION GUIDE* under *ENABLING A GLOBAL USER EXIT* for an explanation of these codes.

WNDO0015. DFHPCP MODULE IS IN SVA/LPA OR ALTERED BY ANOTHER PRODUCT

The DFHPCPxx module can not reside in the SVA or LPA. See CICS REQUIREMENTS in section 10 - INSTALLATION, for further explanation.

ACTION: Remove the DFHPCP module from the DOS SVA or MVS LPA using the procedure described in *CICS-REQUIREMENTS* and bring CICS down and up.

WNDO0016. GUARDIAN INCOMPATIBILITY, CALL UNICOM SYSTEMS FOR PTF.

This message indicates that the release of GUARDIAN that you are using is incompatible with CICS-WINDOWS.

ACTION: Notify Unicom Systems Technical Support (see Appendix C).

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO0017. xxxxxxxx RECORD SIZE IS TOO SMALL

The record size (RECSZ) operand in the IDCAMS DEFINE for the VSAM file xxxxxxxx must be at least 5100.

ACTION: Redefine the VSAM file with the proper record size, then bring CICS down and back up.

WNDO0018. CICS-WINDOWS CURRENTLY UNAVAILABLE

A WNDO,STOP command has been issued to stop all users from doing any WNDO transactions until some sort of system activity has been completed. No WNDO transactions will be accepted until a WNDO,START command is issued.

Note that if you are trying to do a WNDO,START command after doing a WNDO,BACKOUT command, you must do a CEMT NEWCOPY command on the WINDOWS and WNDOMAIN modules.

WNDO0019. JUGL,BACKOUT MUST BE DONE TO START WNDO

The CICS-JUGGLER package has been used before any WNDO transaction was issued. The two packages can not both run at the same time. If you want to activate WINDOWS, you must do a JUGL,BACKOUT command.

ACTION: Make all CICS-JUGGLER users do a JUGL,OFF (or purge all users), then do a JUGL,BACKOUT command, then do any WNDO transaction.

WNDO0020. MULTIPLE INTERACTIVE INTERFACE SESSIONS NOT ALLOWED

This message occurs when, after toggling into a virtual terminal other than session number 1, the operator pressed PF3 to return to the Interactive Interface selection panel and multiple Interactive Interface support has not been specified for this terminal (VSE INTERACTIVE INTERFACE=NO in the User Options Table). The operator must start applications using transaction codes in all sessions except number 1.

ACTION: If you desire to support multiple Interactive Interface sessions for this terminal, refer to *RUNNING CICS-WINDOWS WITH THE DOS VSE INTERACTIVE INTERFACE* in section 13 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS* for the coding required in the User Option Table to support this feature.

WNDO0021. CANNOT ACTIVATE, UNABLE TO CONSTRUCT WNDOTBL

During CICS-WINDOWS activation, CICS-WINDOWS was not able to build an internal image of the customization table and was unable to activate.

ACTION: This message should not occur, however if it does, perform a NEWCOPY of all CICS-WINDOWS programs, then try to activate CICS-WINDOWS. If the problem still occurs, call Unicom Systems Technical Support.

WNDO0022. xxxx CAN'T BE STARTED WITH CICS-WINDOWS

The transaction code represented by xxxx in the message is present in the Transaction Stop table in the User option table, which means that this transaction can not be issued on this terminal if CICS-WINDOWS is active.

ACTION: You must do a WNDO,OFF command in order to use this transaction code.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO0023. YOU MUST END CONVERSATIONAL MODE, PRESS ENTER TO RESTORE

An attempt to toggle to another session has failed because you were trying to toggle out of a conversation task.

ACTION: You must end conversational mode before toggling.

WNDO0024. WNDO,BACKOUT MUST BE DONE TO USE xxxx

The transaction code represented by xxxx in the message is present in the Transaction Backout table. This means that this transaction can not be issued on any terminal if CICS-WINDOWS is active on any terminal.

ACTION: You must do a WNDO,BACKOUT command in order to use this transaction code.

WNDO0025. WNDOMAIN PROGRAM NOT FOUND, CANNOT ACTIVATE

CICS-WINDOWS could not find the WNDOMAIN program. Either the PPT entry is disabled or not present, or WNDOMAIN is not in a CICS load file.

ACTION: Ensure that the WNDOMAIN program is available to CICS-WINDOWS.

WNDO0026. FILE 'xxxxxxxx' STRNO IS LESS THAN 3, CANNOT USE

This message can occur at CICS-WINDOWS initialization if the VSAM file in use has an inadequate number of strings (STRNO) defined. The STRNO value in the FCT entry must be at least '3'. CICS-WINDOWS is unable to use the file as defined, and therefore is unable to activate.

ACTION: In order to use this VSAM file, the STRNO value in the FCT must be increased to 3 or greater. You can not do any WNDO command until this is corrected.

WNDO0027. FILE 'xxxxxxxx' RECFORM MUST BE VARIABLE, CANNOT USE

This message can occur at CICS-WINDOWS initialization if the VSAM file in use has an incorrect record format (RECFORM) defined. The RECFORM value in the FCT entry must be 'VARIABLE'. CICS-WINDOWS is unable to use the file as defined, and therefore is unable to activate.

ACTION: In order to use this VSAM file, the RECFORM parameter in the FCT must be changed to 'VARIABLE'. You can not do any WNDO command until this is corrected.

WNDO0028. PROGRAM 'WNDOENAB' NOT IN PPT OR NOT IN CICS LOAD FILE

CICS-WINDOWS could not find the WNDOENAB program. Either the PPT entry is disabled or not present, or WNDOENAB is not in a CICS load file.

ACTION: Ensure that the WNDOENAB program is available to CICS-WINDOWS.

WNDO0029. YOU MUST SUCCESSFULLY PERFORM A SIGNON BEFORE PROCEEDING

The action that you have selected requires that you be signed-on.

ACTION: Sign on.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO0030. PROGRAM WNDOAUXL HAS NOT BEEN INSTALLED, UNABLE TO INITIALIZE

CICS-WINDOWS could not find the program "WNDOAUXL". Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WNDO0031. TRANSACTION xxxx MUST HAVE A TWASIZE OF AT LEAST 32 BYTES

The PCT entry for the transaction indicated by xxxx must have a TWA size of at least 32 bytes.

ACTION: Using RDO or the DFHPCT macro, increase the TWA size.

WNDO0052. CANNOT START INTERACTIVE INTERFACE SESSION
WNDO0053. CANNOT START INTERACTIVE INTERFACE SESSION
WNDO0054. CANNOT START INTERACTIVE INTERFACE SESSION
WNDO0055. CANNOT START INTERACTIVE INTERFACE SESSION
WNDO0056. CANNOT START INTERACTIVE INTERFACE SESSION
WNDO0057. CANNOT START INTERACTIVE INTERFACE SESSION
WNDO0058. CANNOT START INTERACTIVE INTERFACE SESSION

All seven of the above listed messages contain the same message text. The message number defines the condition which was encountered.

These messages originated in the WNDOVSP program which controls the method of operation when using multiple Interactive Interface selection panels in a DOS VSE environment. In all cases, the message is produced for terminals with VSE INTERACTIVE INTERFACE=YES or VSE INTERACTIVE INTERFACE=SIGNON in the User Option Table when a WNDO,ON command is issued and it is not possible to start multiple Interactive Interface sessions for some reason.

Following is an explanation of the condition that was encountered according to the message number:

WNDO0052	-	Unable to locate the IESO transaction work area.
WNDO0053	-	Unable to locate the system vector.
WNDO0054	-	The terminal is not present in the SP anchor table.
WNDO0055	-	No user status record present.
WNDO0056	-	Unrecognized user status record.
WNDO0057	-	Unable to locate panel hierarchy.
WNDO0058	-	Unable to locate the IESO task.

ACTION: This condition could occur if you are not running DOS VSE but have installed the WNDOVSP program. WNDOVSP only pertains to VSE systems. If this is the case, remove WNDOVSP from the system.

It could also occur if you have activated CICS-WINDOWS using a different Pseudo Terminal ID than that which is coded in the Auto-Init table entry for this terminal. If you used WNDO,INIT, for example and entered a different Pseudo ID or entered no Pseudo IDs.

If either of the above are true, the error will not impact CICS-WINDOWS operation but you will not be able to start multiple Interactive Interface sessions on this terminal.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

If neither of the above is true, call Unicom Systems Technical Support and report the message number to a technical support representative.

WNDO0059. WNDOVSP SUCCESSFULLY ACTIVATED

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the WNDOVSP program is installed. It indicates that support for multiple Interactive Interface sessions has been successfully established.

ACTION: No action is required.

WNDO0060. USER OPTION TABLE MUST BE PRESENT TO ACTIVATE WNDOVSP

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the WNDOVSP program is installed. It indicates that the User Option Table, WNDOTBL, could not be found.

ACTION: If you intend to support multiple Interactive Interface sessions, refer to *RUNNING CICS-WINDOWS WITH THE DOS VSE INTERACTIVE INTERFACE* in section 13 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS* for the coding required in the User Option Table to support this feature. Otherwise, remove WNDOVSP from the system.

WNDO0061. MULTIPLE INTERACTIVE INTERFACE SUPPORT IS NOT ACTIVE

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the WNDOVSP program is installed. It indicates that support for multiple Interactive Interface sessions could not be established for one of the following reasons:

1). All terminals in the User Option Table specify VSE INTERACTIVE INTERFACE=NO.

2). No Auto-Init table entries are present.

ACTION: If you intend to support multiple Interactive Interface sessions, refer to *RUNNING CICS-WINDOWS WITH THE DOS VSE INTERACTIVE INTERFACE* in section 13 - *UNIQUE ENVIRONMENTS AND SPECIAL SITUATIONS* for the coding required in the User Option Table to support this feature. Otherwise, remove WNDOVSP from the system.

WNDO0062 .NON-EXISTENT TERMINAL IN AUTOTBL, ID=xxxx

This message is displayed on the system console in a DOS VSE environment at CICS start-up if the WNDOVSP program is installed. It indicates that an Auto-Init table entry was found which specifies a terminal ID which is not defined to the CICS system. In the message, xxxx is the terminal ID in question.

ACTION: No action is required, the invalid terminal ID will be ignored.

WNDO0063. CORRUPTED CHAIN DETECTED, CD=xx, SNAP DUMP TAKEN

This is a should-not-occur message, indicating that CICS-WINDOWS has detected a storage violation in its internal control blocks.

ACTION: Contact Unicom Systems Technical Support (818) 838-0606. He will probably ask you to print the dump and send it to him.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO1201. COPY FUNCTION COMPLETED

In response to a copy function, this message appears to confirm that the copy function was completed successfully.

ACTION: No action is required.

WNDO1202. NOTHING TO PASTE

In response to a paste function, CICS-WINDOWS was unable to perform the paste because no previous copy function was completed.

ACTION: You must first perform a copy function.

WNDO1401. MAXIMUM NUMBER OF TERMINALS HAS BEEN REACHED.

This message can occur if more terminals are using CICS-WINDOWS then the available dynamic storage will allow or if the User Option table is coded to limit the number of active users and that limit has been reached.

ACTION: Terminate CICS-WINDOWS on some other terminal and try again.

WNDO1402. xxxx IS ALREADY IN USE.

This message appears when attempting to activate Pseudo Terminals and the terminal ID which was entered is either a real physical terminal ID known to CICS or is a Pseudo Terminal ID in use on this or some other terminal.

ACTION: Enter another code for the Pseudo Terminal ID. Note that you may not duplicate codes.

WNDO1403. THIS TERMINAL CAN NOT USE CICS-WINDOWS

This terminal has been excluded from using CICS-WINDOWS in the User Option Table.

ACTION: Notify your supervisor or use another terminal.

WNDO1404. THIS OPERATOR CAN NOT USE CICS-WINDOWS

The 3-character Operator ID assigned to this operator sign-on has been excluded from using CICS-WINDOWS in the User Option Table.

ACTION: Notify your supervisor or use another operator sign-on.

WNDO1405. AUTO-START TABLE REQUIRED BUT NOT PRESENT

AUTOTBL=FORCE has been specified in the User Option Table but no valid TYPE=AUTOTBL statement could be found.

ACTION: Correct and re-assemble the User Option Table and retry.

WNDO1406. NUMBER OF TERMINALS MUST BE FROM 2 TO x.

During initialization of CICS-WINDOWS at a specific physical terminal, in response to the query "SELECT NUMBER OF TERMINALS", the operator entered a number outside the range of 2 to x (where x is the number coded for MAX LOGICAL TERMINALS in the User Options Table).

ACTION: Enter the number again.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO1407. ENTER IS AN INVALID TOGGLE/WINDOW KEY.

In response to the query "SELECT TOGGLE KEY", the operator pressed the ENTER key instead of a Program Function (PF) or Program Attention (PA) key.

ACTION: Press any PF or PA key.

WNDO1408. ERROR RETURN FROM TEMP-STORAGE IO, CODE=xx.

A non-zero code was returned by CICS from a Temporary Storage data access. This problem can occur if a CICS task running under CICS-WINDOWS causes a storage violation, or if auxiliary Temporary Storage is in use and the control interval size of the Temporary Storage dataset is smaller than the largest terminal screen size plus 500 bytes.

ACTION: If problem persists, notify Unicom Systems.

WNDO1409. ACTIVE TASK xxxx ON LOGICAL TERMINAL n MUST BE ENDED.

The operator has attempted to terminate CICS-WINDOWS while a conversational task is waiting for a response in another logical terminal. xxxx is the task ID, n is the logical terminal number. If the "Clear all terminals before logoff" option is specified in the user option table, the message will appear if any logical terminal contains a screen display at the time WNDO,OFF is entered.

ACTION: return to the indicated logical terminal (n) and respond to the task waiting to be ended. Then enter WNDO,OFF again to terminate CICS-WINDOWS.

WNDO1410. PF/PA KEY IS CURRENT TOGGLE KEY
WNDO1410. PF/PA KEY IS CURRENT WINDOW KEY

In response to the query "SELECT WINDOW KEY", the operator pressed the same PF or PA key used for the "toggle" key.

ACTION: Select another PF or PA key.

WNDO1411. CICS-WINDOWS IS CURRENTLY UNAVAILABLE

A WNDO,STOP command has been issued to prevent anyone from using CICS-WINDOWS. The WNDO,ON or WNDO,INIT command can not be performed until a WNDO,START is issued.

ACTION: Wait until CICS-WINDOWS has been made available and try again.

WNDO1412. ID xxxx FOR TERMINAL yyyy ALREADY IN USE

During CICS-WINDOWS initialization when using the PLT Start-up option, the Pseudo Terminal ID indicated by xxxx in the message is defined in the Auto-Start Table for terminal yyyy. Another terminal in the Auto-Start table has defined the same Pseudo Terminal ID.

ACTION: CICS-WINDOWS will not be started on this terminal. Initialization will continue with the next terminal in the Auto-Start table.

WNDO1413. ERROR IN PROCESSING AUTO START ENTRY FOR TERMINAL xxxx

During CICS-WINDOWS initialization when using the PLT Start-up option, a coding error has been found in the Auto-Start Table for terminal xxxx. The previous error message on the console indicates the error condition.

ACTION: CICS-WINDOWS will not be started on this terminal. Initialization will continue with the next terminal in the Auto-Start table.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO1414. ** CICS-WINDOWS WILL EXPIRE IN xx DAYS **

The trial period for CICS-WINDOWS is about to expire. The xx in the message indicates the number of days until it will be automatically deactivated.

ACTION: Notify the MIS System Programmer or call Unicom Systems.

WNDO1415. AUTOTBL KEY CONFLICTS WITH PRINT KEY - xxxx

The Toggle-Forward, Toggle-Backward or Direct Session key in the User Profile designated by xxxx in the message is the same key defined to CICS as the terminal print key (for printing terminal screens on an on-line printer).

This is an informational message indicating that the designated key could not be activated due to the conflict.

ACTION: Change the key in the Auto-Start table to another PF or PA key, or change the SITPRINT operand in the CICS System Initialization Table to specify a different PRINT key.

WNDO1416. xxxxxx KEY CONFLICTS WITH SYSTEM PRINT KEY

The PF or PA key pressed for the TOGGLE key or the WINDOW key is the same key defined to CICS as the terminal print key (for printing terminal screens on an on-line printer). This key can not be used for the TOGGLE key.

ACTION: Select another PF/PA key, or change the SITPRINT operand in the CICS System Initialization Table to specify a different PRINT key.

WNDO1417. USER NOT AUTHORIZED FOR THIS COMMAND

If any of the following are coded YES on the User Option table, then the command(s) cannot be entered with the WNDO transaction code:

BACKOUT, CPROFF, CPRON, DEBUG, INIT, PURGE, START, STOP, SYS, TEMP, MAIN and WIN=

You may use the command(s) with the WNSC transaction code if you are authorized for the transaction.

ACTION: Try the command using WNSC. If CICS will not let you issue that transaction, you cannot issue this command.

WNDO1418. INVALID WINDOW COMMAND

The transaction command following the WNDO transaction code is not a valid CICS-WINDOWS command.

ACTION: Refer to Appendix A for a list of valid transaction commands.

WNDO1419. CICS-WINDOWS xxxxxx, TERMINAL yyyy, USER zzz, TIME hh:mm:ss

This message appears in the CICS statistics log each time a WNDO command is issued at any terminal. MESSAGE LOG=xxxx (where xxx is the destination ID) must be specified in the User Option Table to obtain these messages.

In the message, xxxxxx is the command (ON, OFF, PURGE, etc.), yyyy is the terminal ID where the command was given, zzz is the Operator ID at that terminal and hh:mm:ss is the current time of day.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ACTION: No action is required, the message log is for informational purposes only.

WNDO1420. "WIN=" COMMAND NOT VALID FOR POPUP WINDOWS.

The "WIN=" command may only be used for Standard or Variable window modes.

ACTION: Popup window mode allows you to configure the windows however you wish.

WNDO1421. "xxxx" IS NOT A VALID PROFILE ID.

In response to the WNDO,ON,xxxx command, the command was not able to complete properly because the profile xxxx does not exist.

ACTION: Re-enter the command with a valid profile ID.

WNDO1501. ALL USERS MUST WNDO, OFF TO ISSUE THIS COMMAND.

In order to issue any of the following special purpose transaction commands there must not be any active terminal using CICS-WINDOWS:

WNDO,TEMP
WNDO,MAIN
WNDO,BACKOUT

ACTION: Make all terminal operations issue a WNDO,OFF command. You may use the WNDO,STOP as command to prevent anyone from doing and the WNDO,ON on and/or you may use the WNDO,PURGE or PURGE,ALL commands to force users off. (See section 14 - SPECIAL-PURPOSE COMMANDS for further information).

WNDO1502. VSAM FILE 'xxxxxxxx' NOT FOUND IN FCT.

The WNDO,VSAM command has been issued and WNDOFIL has not been defined in the File Control Table.

ACTION: Define the WNDOFIL FCT entry as described in the installation process. If Unicom System's product CICS-FCTD is present you may use it to dynamically define the file without recycling CICS.

WNDO1503. CICS-WINDOWS NOW USING TEMP-STORAGE
WNDO1503. CICS-WINDOWS NOW USING MAIN-STORAGE

This is a confirmation message in response to the WNDO,TEMP or WNDO,MAIN command indicating successful completion.

ACTION: Proceed with any CICS activity.

WNDO1504. TERMINAL xxxx CAN NOT BE LOCATED.

In response to the WNDO,PURGE, xxxx COMMAND. The terminal ID (real or pseudo) designated by xxxx could not be found.

ACTION: Re-enter the command with a valid terminal ID. This may be a real ID in the TCT or a pseudo ID of an active CICS-WINDOWS user.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO1505. PURGE FOR TERMINAL xxxx SUCCESSFUL.

This is a confirmation message in response to the WNDO,PURGE command indicating successful completion. The designated terminal has been removed from the active users of CICS-WINDOWS. This is equivalent to the operator of that terminal doing a WNDO,OFF command.

ACTION: Proceed with any CICS activity.

WNDO1506. TERMINAL xxxx IS NOT USING WINDOWS.

In response to the WNDO,PURGE, xxxx command, the terminal designated by xxxx is not an active CICS-WINDOWS user.

ACTION: There is no need to purge this user. Proceed with any CICS activity.

WNDO1507. CICS-WINDOWS SUCCESSFULLY QUIESCED.

In response to the WNDO,STOP command, this message indicates that no one will be allowed to activate CICS-WINDOWS on their terminal until a WNDO,START command is issued. Users that are already activated will not be affected.

ACTION: Proceed with whatever system action required the WNDO,STOP command, then issue a WNDO,START command.

WNDO1508. CICS-WINDOWS MAY NOW BE NEW-COPIED.

This is a confirmation message in response to the WNDO,BACKOUT command. CICS-WINDOWS has removed its intercept points in CICS and removed the "in-use" condition.

ACTION: You may now issue a CEMT NEWCOPY command for the module "WINDOWS" or "WNDOTBL" or both, if desired. Note that if you previously issued a WNDO,STOP command, you must perform a NEWCOPY on the WINDOWS module. Message WNDO0040 will result when attempting to perform a WNDO,START command after doing the BACKOUT command.

WNDO1509. CICS-WINDOWS SUCCESSFULLY RESTARTED

In response to the WNDO,START command, this message indicates that WNDO,ON or WNDO,INIT commands will now be honored.

ACTION: Proceed with any CICS activity.

WNDO1510. CICS-WINDOWS IS CURRENTLY UNAVAILABLE

A WNDO,STOP command has been issued to prevent anyone from using CICS-WINDOWS. The WNDO,ON or WNDO,INIT command can not be performed until a WNDO,START is issued.

ACTION: Wait until CICS-WINDOWS has been made available and try again.

WNDO1511. PURGE FOR ALL TERMINALS SUCCESSFUL

This is a confirmation message in response to the WNDO,PURGE,ALL command indicating successful completion. All terminals have been removed from the active users of CICS-WINDOWS.

ACTION: Proceed with any CICS activity.

WNDO1512. DEBUG FOR TERMINAL xxxx NOW ON
WNDO1512. DEBUG FOR TERMINAL xxxx NOW OFF

This a confirmation message in response to the WNDO,DEBUG command indicating successful completion of the ON or OFF specification for the designated terminal.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

ACTION: If you just turned DEBUG on, CICS-WINDOWS will begin writing transaction dumps to the Dump Dataset for every WNDO transaction until a DEBUG OFF command is issued. This will fill up the Dump Dataset if you leave it on very long.

WNDO1513. ALL SESSIONS HAVE BEEN CLEARED

This message appears any time the CLEAR key is depressed in full-screen mode if CLEAR ALL SESSIONS ON CLEAR has been selected in the User Option Table. It indicates that the transactions in all virtual terminals for this physical terminal have been terminated and their screens erased.

ACTION: Continue with any CICS activity. Note that you cannot get rid of this message by pressing CLEAR again. You may start another transaction by entering the transaction code and pressing ERASE EOF before pressing ENTER.

WNDO1514. UNABLE TO PURGE xxxx, ICCF IS ACTIVE

This message can occur when a WNDO,PURGE command is issued, either for a specific terminal or for all terminals. In the message, xxxx will be the specific terminal ID or the work 'ALL'.

The PURGE could not be accomplished because ICCF is active in a virtual terminal other than the current session at that terminal. CICS-WINDOWS will not purge a terminal if ICCF is active.

ACTION: If the PURGE was for ALL terminals, press ENTER at this point and you will be presented with the WNDO,SYS display. The first terminal on the display is the terminal with ICCF active. All other terminals down to that one have been purged.

You can follow the terminal ID or the word 'ALL' in the PURGE command with the word FORCE, if desired. This will go ahead and purge the terminal even though ICCF is active. In DOS VSE systems, this causes an outstanding ICCF session which can not be restarted. You will not be able to log on to ICCF with that Operator ID until CICS is recycled. For non-SP systems, the next time ICCF is entered on that terminal you will receive an ICCF forced-logoff message.

If you do not want to FORCE the PURGE, terminate ICCF on the terminal normally, then re-issue the PURGE command, if needed.

WNDO1515. TASK xxxx PURGED DUE TO WNDO,TIMEOUT

This message occurs when a conversational transaction has been automatically purged by CICS-WINDOWS. Conversational transactions will be purged if the TIME OUT SELECTION TYPE option is specified in the User Option Table and the transaction indicated by xxxx in the message has been inactive for the specified time interval. This message replaces the session display in the virtual terminal where transaction xxxx was initiated, and will be seen when the operator toggles into that virtual terminal.

ACTION: No action is required. You may start the same or any other transaction in this virtual terminal, if desired.

WNDO1516. DATA COMPRESSION IS NOW ON
WNDO1516. DATA COMPRESSION IS NOW OFF

This message occurs in response to the WNDO,CPRON or CPROFF commands. The message indicates that the action was successful and the Data Compression feature of CICS-WINDOWS is now either ON or OFF accordingly.

ACTION: No action is required.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

WNDO1517. INVALID WINDOW COMMAND

The transaction command following the WNDO transaction code is not a valid CICS-WINDOWS command.

ACTION: Refer to Appendix A for a list of valid transaction commands.

WNDO1518. PSEUDO TERMINAL IDS ARE NOW ON
WNDO1518. PSEUDO TERMINAL IDS ARE NOW OFF

This message occurs in response to the WNDO,ALT command. The message indicates that the action was successful and the Pseudo Terminal ID feature of CICS-WINDOWS is now either ON or OFF.

ACTION: No action is required.

WNDO1520. CICS-WINDOWS AUTO-PURGED,

If FORCE PURGE AT SIGNON AND SIGNOFF is specified in the User Option Table, this message will appear any time a sign-on is performed when CICS-WINDOWS is active at the terminal. The message indicates that CICS-WINDOWS has been automatically purged. All sessions have been terminated and any conversational tasks abnormally terminated.

ACTION: No action is required.

WNDO1521. PROGRAM "xxxxxxxx" IS NOT AVAILABLE

CICS-WINDOWS could not find the program xxxxxxxx. Either the PPT entry is disabled or not present, or the program is not in a CICS load file.

ACTION: Ensure that the program is available to CICS-WINDOWS.

WNDO8001. CURSOR IMPROPERLY POSITIONED

When using the VIEW function of the WNDO,SYS display, the cursor was not positioned to the first character of the transaction code in the terminal session to be displayed.

ACTION: Re-position the cursor by using the TAB or NEW LINE key, then press ENTER again.

WNDO8002. INCOMPATIBLE SCREEN SIZE

When using the VIEW function of the WNDO,SYS display, the screen display in the terminal session to be displayed is larger than the screen of your terminal. This indicates that the object session is using alternate screen size on a different model terminal than yours.

ACTION: This terminal session screen can not be viewed on your terminal. You must go to a terminal with the same or larger screen size.

WNDO8003. TERMINAL IS NOT ACTIVE

The terminal session designated by the cursor position on the WNDO,SYS display is not currently an active CICS-WINDOWS session. There are two possible causes for this situation:

1). The cursor was positioned on a virtual terminal number greater than the maximum virtual terminals in use by that physical terminal (selecting session 5 when the terminal only has four virtual terminals, for instance).

2). Between the time that you did the WNDO,SYS command, then positioned the cursor and pressed ENTER, the operator at that terminal did a WNDO,OFF command.

ACTION: Press ENTER to refresh the SYS display. If the terminal still shows on the display, the problem is caused by number 1, above. Re-position the cursor to an active session to try again.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

APPENDIX A - SUMMARY OF CICS-WINDOWS COMMANDS

The types of commands in CICS-WINDOWS are:

API commands	(A)pplication (P)rogram (I)nterface commands are issued from a program to integrate transactions with CICS-WINDOWS.
Control Character commands	These are commands that are preceded with the Control Character. These may be entered from any display in CICS, and will be intercepted by CICS-WINDOWS.
Transaction commands	These are preceded by the WNDO transaction code and must be entered in the upper left-hand corner of the screen.
Window Mode commands	These are entered while in window mode, in the command area above the window.

COMMAND	MODE	DESCRIPTION
A	Window	Designate this window as the active window.
AWINx	API	Designate window number x as the active window.
BACKOUT	Transaction API	Remove CICS-WINDOWS from the system for new copy.
C	Window	Clear this window.
~C	Control Char.	Copy for cut and paste procedure.
CPROFF	Transaction	Deactivate data compression.
CPRON	Transaction	Activate data compression.
CWINx	API	Clear window number x.
DEBUG,xxxx,OFF	Transaction	Turn off DEBUG mode for terminal xxxx.
DEBUG,xxxx,ON	Transaction	Turn on DEBUG mode for terminal xxxx.
Dnn	Window	Scroll this window down nn rows.
E	Window	Pull down Exit Menu.
EXPNx	API	Expand window number x to full screen mode.
H	Window	Pull down Help Menu.
~H	Control Char.	Invoke help. Same as pressing the Help key in an unprotected field.
HELP	Transaction	Invoke CICS-WINDOWS Help Index.
INIT	Transaction	Activate CICS-WINDOWS on a terminal, bypassing the Auto-Start Table.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INQ	Transaction	Display User Configuration Screen. Same as WNDO with no operands.
K	Window	Popup the KEYS window, showing all hot key allocations.
~K	Control Char.	Popup the KEYS window, showing all hot key allocations.
Lnn	Window	Scroll this window left nn columns.
M	Window	Popup the Control Window.
~M	Control Char.	Popup the Control Window.
MAIN	Transaction	Switch from using Temporary Storage to using MAIN storage for saving screen data.
~MSG	Control Char.	Receive message. Valid for the CICS-WINDOWS Message Broadcasting Facility only.
Nx	Window	Pull down the designated action bar pull-down menu. Where x is 1, 2, 3, etc. Note that instead of this method, you may also enter the first character of the name of the menu.
OFF	Transaction API	Terminate CICS-WINDOWS on this terminal.
ON	Transaction API	Activate CICS-WINDOWS on this terminal (using the Auto-Init table).
~P	Control Char.	Paste for cut and paste procedure.
PAN,Wx,Ryy,Czz	API	Set panning (scrolling) position for window x to row yy and column zz.
PURGE,ALL	Transaction API	Remove all terminals from using CICS-WINDOWS.
PURGE,xxxx	Transaction API	Remove a terminal xxxx from CICS-WINDOWS use.
Pxxx	Window	Enter a PF or PA key (xxx can be F1, F2...F24 or A1, A2, A3) This command can be used to resolve PF key conflicts.
Rnn	Window	Scroll this window right nn columns.
RWINx	API	Designate window number x as the receiving window for the next output display.
~S	Control Char.	Stack (copy the selected data onto the end of the clipboard) for the cut and paste procedure.
ST	Window	Popup the Control Window to select a session for toggle.
~ST	Control Char.	Popup the Control Window to select a session for toggle.
SW	Window	Popup the Control Window to select a window to switch to.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

~SW	Control Char.	Popup the Control Window to select a window to switch to.
START	Transaction API	Restart CICS-WINDOWS operation after a previously issued STOP command.
STOP	Transaction API	Stop anyone from performing a WNDO,ON or WNDO,INIT command. Active users will not be affected.
SYS	Transaction	Display the System Screen showing all users.
~TB	Control Char.	Toggle Backward to the next lower numbered session.
TB	API	Toggle Backward to the next lower numbered session.
TEMP	Transaction	Switch from using MAIN storage to Temporary Storage for saving screen data.
~TF	Control Char.	Toggle Forward to the next higher numbered session.
TF	API	Toggle Forward to the next higher numbered session.
~Tn	Control Char.	Toggle directly to session n.
Unn	Window	Scroll this window up nn rows.
~UNP	Control Char.	Unprotect all fields on the screen for following cut and paste procedure.
W	Window	Pull down the Window Menu.
WDnn	Window	Change window height by moving window bar down. For popup window, move the entire window down.
WHnn	Window	Make popup window shorter nn rows.
~WIN	Control Char.	Enter or exit Window Mode.
WIN,IN	API	Enter Window Mode.
WIN,OUT	API	Exit Window Mode.
WIN=n	Transaction	Change the number of windows to n. Valid entries for n are 2, 2H, 2V, 3, or 4. Does not apply to popup window mode.
WLnn	Window	Change window width by moving window bar left. For popup window, move the entire window left.
WNnn	Window	Make popup window narrower nn characters.
WRnn	Window	Change window width by moving window bar right. For popup window, move the entire window right.
WSB	Window	Switch backward to the next lower numbered window
~WSB	Control Char.	Enter window mode (if in Full-screen mode), and switch backward.
WSF	Window	Switch forward to the next higher numbered window.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

~WSF	Control Char.	Enter window mode (if in Full-screen mode), and switch forward.
WSn	Window	Switch to window n as the Dominant Window.
WSnn	Window	Make popup window shorter nn characters.
WTnn	Window	Make popup window taller nn characters.
WUnn	Window	Change window height by moving window bar up. For popup window, move the entire window up.
WWnn	Window	Make popup window wider nn characters.
X	Window	Expand a window into full screen mode.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

APPENDIX B - SYSTEM REQUIREMENTS

This appendix describes the system environment in which CICS-WINDOWS will operate, the program modules involved, the CICS table entries required and the fixed and dynamic storage utilization involved when CICS-WINDOWS is active.

OPERATING SYSTEM:

DOS/VSE, CICS 1.6 and above
OS/MVS, CICS 1.7 and above

TERMINAL TYPES:

Any terminal using IBM 3270 data-stream technology. This includes personal computers accessing the mainframe via the IRMA board or some other interface. The terminal may be in a local or remote environment. Both BTAM and VTAM are supported.

PROGRAM MODULES:

Core-Image/Load modules -
WINDOWS, WNDOENAB, WNDOMAIN, WNDOAUXL, WNDOINIT, WNDOVSP, WNDOMENU, WNDOMSG, WNDODEMO, WNDOHELP, WNDOREFM, WNDOZNEP STS0100, STSPASS, STSCORE, HELPDEMO

Source modules -
WNDOACF2, WNDOCSSF, WNDOCSSN, WNDOINT3, WNDOINT4, WNDOINT5, WNDONEPC, WNDONEPM, WNDOPURG, WNDORSDD, WNDORSDM, WNDOTBL

FILES: VSAM file – WNDOFIL.

CICS TABLE ENTRIES:
PCT - WNDO, WNIT, WAUX, WTMO, WNSC, WMSG, WMNU, WNON, WDMO, WHLP, HDMO, STSC.
PPT - WINDOWS, WNDOMAIN, WNDOINIT, WNDOAUXL, WNDOENAB, WNDOVSP (VSE only), WNDOMSG, WNDOMENU, WNDODEMO, WNDOHELP, HELPDEMO, STS0100, STSPASS, STSCORE
FCT - VSAM file WNDOFIL.

STORAGE REQUIREMENTS:

Resident CICS storage -

WINDOWS - 50K

Shared Storage - approximately 300 bytes (varies depending on Auto-Start options) per physical terminal using CICS-WINDOWS. This storage is allocated above the line at WNDO,ON time and freed at WNDO,OFF time.

Dynamic Storage - (Screen size plus 90 bytes) times 2 each time the Toggle key is pressed or the Window key is pressed.

(Screen size plus 90 bytes) times the number of windows each time a transaction is invoked in window mode.

This storage is held for the duration of the WNDO transaction, it is freed before the application transaction is started.

Temporary Storage or MAIN Storage

(Screen size plus 90 bytes) times the total number of virtual terminals in the system at any one time. This is the sum of all virtual terminals of every physical terminal which has CICS-WINDOWS activated. When CICS-WINDOWS is terminated at a physical terminal, the Temp Storage for that terminal is released.

VSAM File space.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

The VSAM control file contains all customization records, plus approximately 1125 internal screen and on-line help records.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

APPENDIX C – INSTALLATION

INSTALLATION OF THE CICS-WINDOWS PRODUCT

This section describes the procedures for installing CICS-WINDOWS in your CICS environment. Procedures for both DOS/VSE and OS/MVS are described.

CICS REQUIREMENTS (*** MUST READ ***)

1) The parameter EXITS=YES must be specified in the System Initialization Table or as a SIT override.

2) The module DFHPCPxx must run in the same partition or region as CICS. That is, for MVS, these modules can not reside in the LPA and for DOS, they can not reside in the SVA. (xx is the CICS Module suffix).

   CICS 3.3 users must either unprotect the extended read-only DSA (RENTPGM=NOPROTECT in DFHSIT) or remove the reentrant flag from the DFHPCP load module, which will cause it to load in storage protect key 8, rather than zero.

3) If your CICS contains a Nucleus Load Table (NLT) with PROTECT=YES, the DFHPCPxx must not be included in the NLT (xx is the CICS Module suffix).

4) For VTAM systems, if the READ BUFFER keyword of the CICS-WINDOWS customization table is coded YES, the maximum TIOAL parameter of the CICS Terminal Control Table may need to be increased. If you experience ATNI abends (particularly on remote terminals), or if your terminal hangs when the Toggle key is pressed, increasing this value will usually correct the problem.

5) CICS-WINDOWS will run on CICS versions 1.7 through 3.3, inclusive.

INSTALLATION STEPS

1. Unload installation library from tape, create installation PDS.
2. Unload CICS-Windows Product
   - 2.1 Unload CICS-Windows load library from tape. It may be necessary to unload multiple load libraries if you are running more than one CICS release at your installation.
   - 2.2. Define and load the CICS-Windows control file.
     - 2.2.1 Optionally migrate any customization options defined in a prior CICS-Windows version to the newly defined CICS-Windows control file.
   - 2.3 Define resources used by CICS-WINDOWS to CICS using RDO.
   - 2.4 Install the resource group with the CEDA transaction (or batch using the DFHCSDUP program). INSTALL GROUP(WNDOGRP) or to install every time CICS is initialized, ADD GROUP(WNDOGRP) LIST(yourlist)
3. Implement DFHZNEP interface. This interface will call WNDOZNEP to deactivate CICS-Windows in the event of a node error condition.
4. Build a product control password module(STSPASS) from the supplied installation codes.
5. Add the CICS-WINDOWS load library to the DFHRPL concatenation of your CICS startup JCL.
6. Optionally code and implement any of the desired sample user exit programs.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

THE INSTALLATION TAPE

The installation tape for VSE installations consists of three files.

FILE1 CSH.WNDO42.VSE.TAPE.FILE1
FILE2 CSH.WNDO42.VSE.TAPE.FILE2
FILE3 CSH.WNDO42.VSE.TAPE.FILE3

The installation tape for MVS installations consists of four files.

FILE1 CSH.WNDO42.INSTLIB
FILE2 CSH.WNDO42.CTS130.LOADLIB (CICS TS 1.3 installation)
FILE3 CSH.WNDO42.CTS220.LOADLIB (CICS TS 2.2 & 2.3 installation)
FILE4 CSH.WNDO42.CONTROL.FILE

DOS/VSE INSTALLATION

Step 1. Load the installation programs and password table

The following JCL can be used to install the first file of the tape for DOS/VSE users:

```
// JOB LOAD STSINST
// ASSGN SYSIPT,xxx      [Note 1]
// MTC REW,SYSIPT
// LIBDEF PHASE,CATALOG=L.S          [Note 2]
// OPTION CATAL
  INCLUDE
// EXEC LNKEDT
// RESET SYSIPT
/*
```

[Notes]:

1. Assign SYSIPT to a tape drive that can read the BPI of your installation tape.
2. This LIBDEF is for DOS/VSE, where
   L.S = Library, Sub-library

The library for the install program, password processor and password table must be a CICS load library.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Step 2. Print installation instructions and punch installation JCL

The following JCL executes the STSINST program to print the installation documentation and punch the install JCL:

```
// ASSGN SYS011,xxx        [Note 1]
// LIBDEF *,SEARCH=L.S  [Note 2]
// EXEC STSINST
PRODUCT=WINDOWS      [Note 3]
MODE=PRINT      [Note 3]
OPSYS=DOS       [Note 3]
CICS=x.x  [Note 3]
LINES=56[Note 3]
JCL=INSTALL      [Note 3]
/*
/&
```

[Notes]:

1. Assign SYS011 to a tape drive that can read the BPI of your installation tape.

2. The LIBDEF must identify the library where STSINST was link-edited as a search library. L.S = Library, Sub-library.

3. See the discussion entitled KEYWORDS OF THE UNICOM INSTALLATION PROGRAM, later in this section, for the meaning of these keywords.

At this point, the full installation instructions will be printed. The installation JCL will be punched into the POWER punch queue. Retrieve or view the printed installation documentation and follow the instructions provided there to complete the installation of CICS-WINDOWS.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

MVS INSTALLATION

CICS-Windows V4.2 Installation Instructions for CICS TS 1.3 and TS 2.2

When going through this document please note that the letters **YYY** represent the numbers **530** for TS 1.3 installations and **620** for TS 2.2 installations. Also note that the letters **ZZZ** represent **130** for TS 1.3 installations and **220** for TS 2.2 installations.

STEP 1

The following JCL will allow you to copy the installation JCL from the CICS-Windows tape which you received. Once you have copied this dataset you will find that it contains member #INST**YYY**.

```
//WN42INS  JOB 1,UNICOM-SYSTEMS,MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//*
//* ====================================================================
//* UNLOAD CICS-WINDOWS 4.2 TRANSACTION SERVER 1.3 or 2.2 INSTALL LIBRARY
//* ====================================================================
//*
//* FILE 1 - INSTALLATION JCLLIB
//*
//UNLOAD   EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN1      DD UNIT=tape,DSN=CSH.WNDO42.INSTLIB,
//            DISP=OLD,LABEL=(1,SL,EXPDT=98000),
//            VOL=(,RETAIN,SER=UNICOM)
//OUT1     DD UNIT=uuuu,DSN=hlq.WNDO42.INSTLIB
//            DISP=(,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=0),
//            SPACE=(TRK,(30,1,20)),VOL=SER=vvvvvv
//SYSIN    DD *
   C I=((IN1,R)),O=OUT1
/*
//
```

Note:
1.You should follow your system naming conventions to replace UNIT=tape with input tape drive name and replace UNIT=uuuu with DASD devices for output datasets and hlq for your authorized high-level qualifier.

2.You should replace VOL=SER=vvvvvv with a suitable volume serial number.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 2

Modify the JCL in member #INST**YYY** to complete the installation process. This JCL will unload the load modules, create a new Windows control file and create resource definitions using the CSD utility program. You can optionally migrate your records from your previous Windows control file.

```
//#INSTALL JOB 1,UNI,MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//* ====================================================================
//*
//* INSTALL CICS-WINDOWS 4.2 FOR CICS 5.3/TS 1.3 or CICS 6.2/TS 2.2
//*
//* ====================================================================
//* NOTES:
//*
//*  1) CHANGE UNIT, VOLSER AND OUT2 DSN.
//*  2) CHANGE NAME OF VSAM DATASET IF DESIRED.
//*  3) CHANGE SHARE OPTIONS  IF DESIRED.
//*  4) CHANGE VOLUME  INFORMATION TO ALLOW ALLOCATIONS OF THE
//*     VSAM DATASETS AT YOUR INSTALLATION.
//*  5) UNCOMMENT STEP05 TO COPY an old V3.x CONTROL FILE RECORDS TO THE NEW
//*     V4.2 FILE.
//*  6) CHANGE 'CICSYYY.CICS.SDFHLOAD' TO THE IBM DISTRIBUTED CICS LOAD
//*     LIBRARY THAT CONTAINS THE DFHCSDUP MODULE.
//*  7) CHANGE 'CICSYYY.DFHCSD' TO THE CSD DSN.
//*  8) NOTE THAT THE RESOURCE DEFINITIONS ARE ACCESSED FROM THE
//*     SAME OUTPUT PDS AS CODED IN STEP 1.
//* ====================================================================
//*
//* ====================================================================
//* STEP 2 - RESTORE LOAD MODULES TO A LOADLIB PDS
//* ====================================================================
//STEP02   EXEC  PGM=IEBCOPY,REGION=2048K
//IN2      DD UNIT=TAPE,DSN=CSH.WNDO42.CTSZZZ.LOADLIB,  /*NOTE1*/
//            DISP=OLD,LABEL=(2,SL,EXPDT=98000),
//            VOL=(,RETAIN,SER=UNICOM)
//OUT2     DD UNIT=SYSDA,DSN=hlq.WNDO42.CTSZZZ.LOADLIB,
//            DISP=(,CATLG),
//            DCB=(RECFM=U,LRECL=0,BLKSIZE=4000),
//            SPACE=(TRK,(45,1,20)),VOL=SER=VVVVVV      /*NOTE1*/
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
   C I=((IN2,R)),O=OUT2
/*
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 3

```
//* ====================================================================
//* STEP 3 - DEFINE THE VSAM CLUSTER * UPDATE VSAM CONTROL CARDS    */
//* CHECK NOTE2 (VSAM DATASET), NOTE3 (SHARE OPTIONS) & NOTE4 (VOLSER)
//* ====================================================================
//STEP03   EXEC PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
  DEFINE CLUSTER                        -
    (NAME(hlq.WND042.CICSYYY.FILE)      -
     INDEXED                            -
     KEYS(40 0)                         -
     RECSZ(2080 8192)                   -
     SHR(3 4)                           -
     FSPC (10 10)                       -
     VOL(VVVVVV))                       -
   DATA                                 -
     (NAME(hlq.WND042.CICSYYY.FILE.DATA) -
     CISZ(8192)                          -
     TRK(40 5))                          -
   INDEX                                -
    (NAME(hlq.WND042.CICSYYY.FILE.INDEX)  -
    TRK(1 1))
/*
//* ====================================================================
//* STEP 4 - RESTORE THE VSAM CLUSTER
//* ====================================================================
//STEP04    EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//IN3       DD UNIT=TAPE,DSN=CSH.WND042.CONTROL.FILE,
//             DISP=OLD,LABEL=(4,SL,EXPDT=98000),
//             VOL=(,RETAIN,SER=UNICOM)
//WNDOFIL   DD DISP=SHR,DSN=hlq.WND042.CICSYYY.FILE
//SYSIN     DD *
  REPRO INFILE (IN3) OUTFILE (WNDOFIL)
/*
//* ====================================================================
//* STEP 5 - OPTIONALLY COPY THE CUSTOMIZATION OPTIONS FROM THE
//*          WINDOWS 3.x CONTROL FILE
//* ====================================================================
//*STEP05    EXEC PGM=IDCAMS
//*SYSPRINT  DD SYSOUT=*
//*WND03x    DD DISP=SHR,DSN=hlq.WND03x.CONTROL.FILE    /*NOTE5*/
//*WNDOFIL   DD DISP=OLD,DSN=hlq.WND042.CICSYYY.FILE
//*SYSIN     DD *
//*  REPRO INFILE  (WND03x)     -
//*        OUTFILE (WNDOFIL)    -
//*        FROMKEY (O)          -
//*        TOKEY   (O)          -
//*        REPLACE
//*
//* ====================================================================
//* STEP 6 - DEFINE RESOURCES TO CICS
//* ====================================================================
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 6 (continued)

```
//STEP06   EXEC PGM=DFHCSDUP
//SYSPRINT DD SYSOUT=*
//STEPLIB  DD DISP=SHR,DSN=CICSYYY.CICS.SDFHLOAD        /*NOTE6*/
//DFHCSD   DD DISP=SHR,DSN=CICSYYY.DFHCSD               /*NOTE7*/
//SYSIN    DD DISP=SHR,DSN=hlq.WNDO42.INSTLIB (#CSDMYYY) /*NOTE8*/
//
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

STEP 7

REVIEW THE WNDONEPC MEMBER IN THIS PDS AND UPDATE YOUR USER REPLACEABLE DFHZNEP PROGRAM. THIS WAS ONCE AN OPTIONAL STEP, BUT IS NOW REQUIRED.

Defining the DFHZNEP Interface is no longer an optional part of the installation process. Failure to call WNDOZNEP from DFHZNEP may be detrimental to CICS system performance.

```
//SAMPLE JOB 1,UNI,MSGCLASS=X,CLASS=A,NOTIFY=&SYSUID
//* ================================================================
//* SAMPLE ASSEMBLE/LINKEDIT FOR DFHZNEP
//* ================================================================
//* NOTES:
//*  1) CHANGE 'CICSXXX.DFHMAC' TO THE CORRECT CICS MACRO LIBRARY.
//*  2) NOTE THAT THE SAMPLE DFHZNEP SOURCE IS ACCESSED FROM THE
//*     SAME OUTPUT PDS CREATE IN STEP 1 OF THE INSTALL JOB.
//*  3) CHANGE 'CICSxxx.CICS.SDFHLOAD' TO THE IBM DISTRIBUTED
//*     CICS LOAD LIBRARY.
//*  4) CHANGE 'CICSxxx.CICS.SDFHLOAD' TO THE IBM DISTRIBUTED
//*     CICS LOAD LIBRARY.
//*
//* ================================================================
//ASM      EXEC PGM=ASMA90,REGION=4096K,
//         PARM='DECK,NOOBJECT,TERM'
//SYSTERM   DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSLIB    DD  DSN=SYS1.MACLIB,DISP=SHR
//          DD  DSN=SYS1.MODGEN,DISP=SHR
//          DD  DSN=CICSxxx.SDFHMAC,DISP=SHR             /*Note 1*/
//SYSUT1    DD  UNIT=VIO,SPACE=(CYL,(5,5))
//SYSPUNCH  DD  DSN=TEMP,UNIT=SYSDA,SPACE=(CYL,(1,1)),
//    DCB=(LRECL=80,BLKSIZE=400,RECFM=FB,DSORG=PS),DISP=(,PASS)
//SYSIN     DD  DSN=hlq.WNDO42.INSTLIB (WNDONEPC),        /*Note 2*/
//          DISP=SHR
//SYSLIN    DD  DUMMY
//LINK     EXEC PGM=IEWL,
//         PARM='LIST,XREF,RENT,LET,',
//         COND=(0,NE)
//SYSPRINT  DD  SYSOUT=*
//SYSUT1    DD  UNIT=VIO,SPACE=(TRK,(5,5))
//SYSLMOD   DD  DSN=CICSxxx.SDFHLOAD(DFHZNEP),DISP=SHR   /*Note 3*/
//SDFHLOAD  DD  DSN=CICSxxx.SDFHLOAD,DISP=SHR            /*Note 4*/
//USERTEXT  DD  DSN=TEMP,DISP=SHR
//SYSLIN    DD  *
 INCLUDE SDFHLOAD(DFHEAI)
 INCLUDE USERTEXT
 INCLUDE SDFHLOAD(DFHEAI0)
 MODE AMODE(31),RMODE(ANY)
 ORDER DFHEAI
 ORDER DFHZNEP0
 ORDER DFHEAI0
 ENTRY DFHZNENA
 NAME DFHZNEP(R)
/*
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

DEFINING THE DFHZNEP INTERFACE

This step MUST be performed in order to operate CICS-WINDOWS; however, prior to using CICS-WINDOWS in a production environment, you should add some instructions to the node error program to purge CICS-WINDOWS from the terminal when an error or time-out occurs.

For details of the DFHZNEP interface, refer to section 15 - *DEACTIVATING CICS-WINDOWS WHEN A NODE ERROR OCCURS.*

ADDITIONAL INSTALLATION NOTES, MVS AND DOS/VSE

1). If you use RDO to define the programs, be sure to specify ASSEMBLER as the language.

2). If you wish to use a different transaction code than WNDO, you may make the TRANSID value any desired transaction code. If more than one transaction code is needed (for separate activity accounting or security reasons), you may provide as many PCT entries as desired, all pointing to program WINDOWS.

3). Slightly improved response time when in window mode may result if the WNDOMAIN program is made resident. It requires about 50K.

4). For MVS users only, if CICS-WINDOWS has been previously installed using the same source PDS, you must first delete the following members from the dataset since IEBUPDTE will not replace the existing members:

WNDOACF2	WNDOCSSF	WNDOCSSN
WNDOINT3	WNDOINT4	WNDOINT5
WNDONEPC	WNDONEPM	WNDOPURG
WNDORSDM	WNDOTBL	

SUMMARY

At this point, the basic CICS-WINDOWS product is installed and ready to operate.

You may want to tailor the product to fit your environmental needs, as described in section 11 - *CUSTOMIZATION*, however the product is operational without doing so.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CICS TABLE ENTRIES FOR CICS RELEASE 1.7 AND 2.1

```
//*
//*       USE THE FOLLOWING TO DEFINE THE PPT, PCT AND FCT ENTRIES.
//*
//STEP06        EXEC   PGM=DFHCSDUP
//SYSPRINT      DD     SYSOUT=*
//STEPLIB       DD     DSN=CICS???.LOADLIB,DISP=SHR       /* NOTE 1 */
//DFHCSD        DD     DSN=CICS???.DFHCSD,DISP=SHR,DISP=SHR
//SYSIN DD      *
DEFINE PROGRAM(STSCORE)

     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STSPASS)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)

DEFINE PROGRAM(STS0100)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WINDOWS)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOAUXL)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOENAB)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOINIT)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOMAIN)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOMENU)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOMSG)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDODEMO)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOHELP)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOZNEP)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(HELPDEMO)
     GROUP(WNDOGRP)
     LANGUAGE(ASSEMBLER)
DEFINE TRANSACTION(STSC)
     GROUP(WNDOGRP)
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    PROGRAM(STSCORE)
    TWASIZE(00032)
DEFINE TRANSACTION(WAUX)
    GROUP(WNDOGRP)
    PROGRAM(WNDOAUXL)
    TWASIZE(00032)
DEFINE TRANSACTION(WMNU)
    GROUP(WNDOGRP)
    PROGRAM(WNDOMENU)
    TWASIZE(00032)
DEFINE TRANSACTION(WMSG)
    GROUP(WNDOGRP)
    PROGRAM(WNDOMSG)
    TWASIZE(00032)
DEFINE TRANSACTION(WNDO)
    GROUP(WNDOGRP)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WNIT)
    GROUP(WNDOGRP)
    PROGRAM(WNDOINIT)
    TWASIZE(00032)
DEFINE TRANSACTION(WNSC)
    GROUP(WNDOGRP)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WTMO)
    GROUP(WNDOGRP)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WNON)
    GROUP(WNDOGRP)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WDMO)
    GROUP(WNDOGRP)
    PROGRAM(WNDODEMO)
    TWASIZE(00032)
DEFINE TRANSACTION(WHLP)
    GROUP(WNDOGRP)
    PROGRAM(WNDOHELP)
    TWASIZE(00032)
DEFINE TRANSACTION(HDMO)
    GROUP(WNDOGRP)
    PROGRAM(HELPDEMO)
    TWASIZE(00032)
/*
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
*************************************************************************
*  4.4    CICS-WINDOWS FCT ENTRY - CICS RELEASE 1.7 AND 2.1
*************************************************************************
DFHFCT  TYPE=DATASET,                                       /* NOTE 2 */ X
          DATASET=WNDOFIL,                                  /* NOTE 2 */ X
          DSNAME=CICS.WINDOWS.CONTROL.FILE,                 X
          DISP=SHR,                                         X
          ACCMETH=(VSAM),                                   X
          FILSTAT=(ENABLED,OPENED),                         X
          RECFORM=(VARIABLE,BLOCKED),                       X
          SERVREQ=(UPDATE,ADD,DELETE,BROWSE,READ),          X
          STRNO=3
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CICS TABLE ENTRIES FOR CICS RELEASE 3.1

```
//*
//*       RUN THE CSD UTILITY TO DEFINE WNDOGRP FOR CICS 3.1
//*
//STEP6 EXEC PGM=DFHCSDUP
//SYSPRINT      DD      SYSOUT=*
//STEPLIB       DD      DSN=CICS311.LOADLIB,DISP=SHR      /* NOTE 1 */
//DFHCSD        DD      DSN=CICS311.DFHCSD,DISP=SHR
//SYSIN DD      *
DEFINE FILE(WNDOFIL)

     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS CONTROL FILE)
     DSNAME(CICS.WINDOWS.CONTROL.FILE)
     LSRPOOLID(1)
     DSNSHARING(ALLREQS)
     STRINGS(003)
     STATUS(ENABLED)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
     DATABUFFERS(00004)
     INDEXBUFFERS(00003)
     RECORDFORMAT(V)

     ADD(YES)
     BROWSE(YES)
     DELETE(YES)
     READ(YES)
     UPDATE(YES)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
DEFINE PROGRAM(STSCORE)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM MEMORY DISPLAY/ALTER)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STSPASS)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM PRODUCT PASSWORD TABLE)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(STS0100)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM PRODUCT PASSWORD CONTROL)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WINDOWS)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS RESIDENT MODULE)
     LANGUAGE(ASSEMBLER)
     RESIDENT(YES)
     CEDF(NO)
DEFINE PROGRAM(WNDOAUXL)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS AUXILIARY FUNCTIONS)
     LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOENAB)
     GROUP(WNDOGRP)
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    DESCRIPTION(CICS-WINDOWS HELP FEATURE)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOINIT)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS INITIALIZATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOMAIN)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MAINLINE)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOMENU)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MENU GENERATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOMSG)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MESSAGE BROADCAST)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDODEMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DEMONSTRATION)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOHELP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP PROCESSOR)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(WNDOZNEP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DFHZNEP INTERFACE)
    LANGUAGE(ASSEMBLER)
DEFINE PROGRAM(HELPDEMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP DEMO DISPLAY)
    LANGUAGE(ASSEMBLER)
DEFINE TRANSACTION(STSC)
    GROUP(WNDOGRP)
    DESCRIPTION(UNICOM MEMORY DISPLAY/ALTER)
    PROGRAM(STSCORE)
    TWASIZE(00032)
DEFINE TRANSACTION(WAUX)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUXILIARY FUNCTIONS)
    PROGRAM(WNDOAUXL)
    TWASIZE(00032)
DEFINE TRANSACTION(WMNU)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MENU GENERATION)
    PROGRAM(WNDOMENU)
    TWASIZE(00032)
DEFINE TRANSACTION(WMSG)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MESSAGE BROADCAST)
    PROGRAM(WNDOMSG)
    TWASIZE(00032)
DEFINE TRANSACTION(WNDO)
    GROUP(WNDOGRP)
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    DESCRIPTION(CICS-WINDOWS PRIMARY TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WNIT)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS STATUS DISPLAY)
    PROGRAM(WNDOINIT)
    TWASIZE(00032)
DEFINE TRANSACTION(WNSC)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS SECURED TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WTMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS TIMEOUT TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WHLP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP DIRECTORY TRANCODE)
    PROGRAM(WNDOHELP)
    TWASIZE(00032)
DEFINE TRANSACTION(WNON)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUTO-ON TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
DEFINE TRANSACTION(WDMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DEMO TRANCODE)
    PROGRAM(WNDODEMO)
    TWASIZE(00032)
DEFINE TRANSACTION(HDMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP DEMO TRANCODE)
    PROGRAM(HELPDEMO)
    TWASIZE(00032) /*
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
//*
//*       RUN THE CSD UTILITY TO DEFINE WNDOGRP FOR CICS 3.2
//*
//STEP6 EXEC   PGM=DFHCSDUP
//SYSPRINT     DD      SYSOUT=*
//STEPLIB      DD      DSN=CICS321.SDFHLOAD,DISP=SHR    /* NOTE 1 */
//SYSIN DD     *
DEFINE FILE(WNDOFIL)

     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS CONTROL FILE)
     DSNAME(CICS.WINDOWS.CONTROL.FILE)
     LSRPOOLID(1)
     DSNSHARING(ALLREQS)
     STRINGS(003)
     STATUS(ENABLED)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
     DATABUFFERS(00004)
     INDEXBUFFERS(00003)
     RECORDFORMAT(V)
     ADD(YES)
     BROWSE(YES)
     DELETE(YES)
     READ(YES)
     UPDATE(YES)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
DEFINE PROGRAM(STSCORE)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM MEMORY DISPLAY/ALTER)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
DEFINE PROGRAM(STSPASS)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM PRODUCT PASSWORD TABLE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
DEFINE PROGRAM(STS0100)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM PRODUCT PASSWORD CONTROL)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
DEFINE PROGRAM(WINDOWS)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS RESIDENT MODULE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
     RESIDENT(YES)
     CEDF(NO)
DEFINE PROGRAM(WNDOAUXL)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS AUXILIARY FUNCTIONS)
     LANGUAGE(ASSEMBLER)
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOENAB)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS EXIT ENABLER)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOINIT)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS INITIALIZATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOMAIN)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MAINLINE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOMENU)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MENU GENERATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOMSG)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MESSAGE BROADCAST)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDODEMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DEMONSTRATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOHELP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP FEATURE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(WNDOZNEP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DFHZNEP INTERFACE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE PROGRAM(HELPDEMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP DEMO TRANCODE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
DEFINE TRANSACTION(STSC)
    GROUP(WNDOGRP)
    DESCRIPTION(UNICOM MEMORY DISPLAY/ALTER)
    PROGRAM(STSCORE)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WAUX)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUXILIARY FUNCTIONS)
    PROGRAM(WNDOAUXL)
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WHLP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP FEATURE)
    PROGRAM(WNDOHELP)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WMNU)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MENU GENERATION)
    PROGRAM(WNDOMENU)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WMSG)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MESSAGE BROADCAST)
    PROGRAM(WNDOMSG)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WNDO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS PRIMARY TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WNIT)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS STATUS DISPLAY)
    PROGRAM(WNDOINIT)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WNSC)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS SECURED TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WTMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS TIMEOUT TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WNON)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUTO-ON TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
DEFINE TRANSACTION(WDMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DEMO TRANCODE)
    PROGRAM(WNDODEMO)
    TWASIZE(00032)
    TASKDATALOC(ANY)
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
DEFINE TRANSACTION(HDMO)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS HELP DEMO TRANCODE)
     PROGRAM(HELPDEMO)
     TWASIZE(00032)
     TASKDATALOC(ANY)
/*
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CICS TABLE ENTRIES FOR CICS RELEASE 3.3

```
//*
//*
//STEP6 EXEC   PGM=DFHCSDUP
//SYSPRINT     DD       SYSOUT=*
//STEPLIB      DD       DSN=CICS330.SDFHLOAD,DISP=SHR    /* NOTE 1 */
//DFHCSD       DD       DSN=CICS330.DFHCSD,DISP=SHR
//SYSIN DD     *
DEFINE FILE(WNDOFIL)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS CONTROL FILE)
     DSNAME(CICS.WINDOWS.CONTROL.FILE)
     LSRPOOLID(1)
     DSNSHARING(ALLREQS)

     STRINGS(003)
     STATUS(ENABLED)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
     DATABUFFERS(00004)
     INDEXBUFFERS(00003)
     RECORDFORMAT(V)
     ADD(YES)
     BROWSE(YES)
     DELETE(YES)
     READ(YES)
     UPDATE(YES)
     OPENTIME(FIRSTREF)
     DISPOSITION(SHARE)
DEFINE PROGRAM(STSCORE)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM MEMORY DISPLAY/ALTER)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
     EXECKEY(CICS)
DEFINE PROGRAM(STSPASS)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM PRODUCT PASSWORD TABLE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
     EXECKEY(CICS)
DEFINE PROGRAM(STS0100)
     GROUP(WNDOGRP)
     DESCRIPTION(UNICOM PRODUCT PASSWORD CONTROL)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(BELOW)
     EXECKEY(CICS)
DEFINE PROGRAM(WINDOWS)
     GROUP(WNDOGRP)
     DESCRIPTION(CICS-WINDOWS RESIDENT MODULE)
     LANGUAGE(ASSEMBLER)
     DATALOCATION(ANY)
     EXECKEY(CICS)
     RESIDENT(YES)
     CEDF(NO)
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
DEFINE PROGRAM(WNDOAUXL)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUXILIARY FUNCTIONS)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOENAB)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS EXIT ENABLER)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOINIT)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS INITIALIZATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOMAIN)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MAINLINE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOMENU)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MENU GENERATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOMSG)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MESSAGE BROADCAST)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDODEMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DEMONSTRATION)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOHELP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP PROCESSOR)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(WNDOZNEP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DFHZNEP INTERFACE)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE PROGRAM(HELPDEMO)
    GROUP(WNDOGRP)
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    DESCRIPTION(CICS-WINDOWS HELP DEMO DISPLAY)
    LANGUAGE(ASSEMBLER)
    DATALOCATION(ANY)
    EXECKEY(CICS)
DEFINE TRANSACTION(STSC)
    GROUP(WNDOGRP)
    DESCRIPTION(UNICOM MEMORY DISPLAY/ALTER)
    PROGRAM(STSCORE)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WAUX)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUXILIARY FUNCTIONS)
    PROGRAM(WNDOAUXL)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WMNU)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MENU GENERATION)
    PROGRAM(WNDOMENU)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WMSG)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS MESSAGE BROADCAST)
    PROGRAM(WNDOMSG)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WNDO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS PRIMARY TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WNIT)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS STATUS DISPLAY)
    PROGRAM(WNDOINIT)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WNSC)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS SECURED TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WTMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS TIMEOUT TRANCODE)
```


© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

```
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WNON)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS AUTO-ON TRANCODE)
    PROGRAM(WINDOWS)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WDMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS DEMO TRANCODE)
    PROGRAM(WNDODEMO)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(WHLP)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP DIRECTORY TRANCODE)
    PROGRAM(WNDOHELP)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
DEFINE TRANSACTION(HDMO)
    GROUP(WNDOGRP)
    DESCRIPTION(CICS-WINDOWS HELP DEMO TRANCODE)
    PROGRAM(HELPDEMO)
    TWASIZE(00032)
    TASKDATALOC(ANY)
    TASKDATAKEY(CICS)
/*
```

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

****
* 3.3 JCL NOTE EXPLANATIONS
****

NOTE 1 Must be the library that "STSINST" phase was link edited to in the initial JCL process from the installation guide.

2 This tape drive should have the mounted master product install tape received from Unicom Systems.

3 These are temporary files that are written to disk and are deleted after use.

4 This is a CICS library that would be used to compile and link-edit a command level program.

5 This is the load library where CICS programs are kept. It must be in the concatenation for DFHRPL in the CICS startup JCL.

6 SYSUT1 and SYSUT2 should point to a source PDS for macros. If the macros already exist, you should delete them before the install/Reinstall process.

7 VOL=(??????) - DASD volume where the WNDOFIL is to reside.

8 WNDOFILE=??????? - Must be the 7-byte VSAM filename.
WNDOFILE=BYPASS - Will bypass the file.

9 MODE=Install for a new installation of the Windows product.
MODE=Reinstall for the re-installation of the Windows product.

10 WNDOFIL=BYPASS|???????

BYPASS = Bypass writing to the WNDOFIL.

?????? = Name of the WNDOFIL for your installation. STSINST will add records to this file or replace existing records depending upon the INSTALL|REINSTALL parameter.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

APPENDIX D - REPORTING PROBLEMS

If a problem is encountered with CICS-WINDOWS which can not be resolved by the information in this document, you may call Unicom Systems at (818) 838-0606. You will be referred to a technical support representative who will take the information about the problem and resolve it then and there on the phone, if possible. Should a support person not be available when you call, you will be called back as soon as possible, usually the same day.

If the problem can not be resolved on the phone, the support representative will require that you run certain CICS diagnostic reports and send them to him. This will usually include an Auxiliary Trace of the transaction or series of transactions involved, a core dump if one is available and a listing of your User Option Table. He may also need the CICS Terminal Control Table.

When you call, please have the following information available:

1) Your Operating System.
2) The release of CICS where CICS-WINDOWS runs.
3) The release and PTF level number of CICS-WINDOWS that you have (This is shown on the User Configuration display).
4) Whether you use BTAM or VTAM.
5) Any message numbers involved in the problem, whether produced by CICS, CICS-WINDOWS or the Operating System.

Unicom Systems is committed to effective product support, and we will make every effort to resolve your problem as quickly as possible.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

GLOSSARY OF TERMS

There are various terms used in this manual and often used by people when talking about the features of the product which, while not totally unique to CICS-WINDOWS, are defined here for better understanding.

Term	Meaning
ABEND	This is a data processing term which means “abnormal end.” It is the condition where a program does not reach a logical end where it either goes away or proceeds to the next program, but rather some convention of programming is violated and CICS is forced to “purge” the task from the system. In CICS, a transaction ABEND will be indicated by an unexpected message to the screen, usually accompanied by a 4-character code, such as ASRA, or APCT, etc.
ACTION BAR	This refers to the area usually at the top of the display, that may be used for invoking functions by tabbing to the PULL-DOWN MENU name and pressing ENTER. Then when the pull-down menu appears, tabbing to the desired option and pressing ENTER to invoke the function.
ACTIVE TASK	This refers to a virtual terminal in which a transaction has been started and that transaction is expecting more input from the operator. An active task is normally conversational or pseudo conversational.
ACTIVE WINDOW	An active window is the window which contains an “A” in the window command area at the top. Transactions can be updated and initiated only in an active window and there can only be one active window at a time.
APPLICATION	This refers to a program or transaction, which is available to run in your CICS environment. An application often consists of several programs or transactions. Thus, we might refer to Data Entry as an application, when several transactions are involved to perform all Data Entry functions.
CONVERSATIONAL	This is a programming technique whereby the program does a “read” on the terminal when it needs more input from the operator. Conversational transactions do not release the CICS resources that they use when they display a screen. The resources are not released until the program comes to a normal or abnormal end.
DOMINANT WINDOW	A dominant window is one which is active (unprotected) while all other windows are protected, and may or may not overlap other windows. When using “variable” or “popup” window modes, there is always a dominant window.
FULL SCREEN	This term is used to designate the normal way that transactions are run. That is, the screen is not divided up into multiple parts, or windows.
FULL SCREEN MODE	The normal mode of operation, where there are no windows on the screen.
HOOKS	A “hook” is a technique used by various vendor products to get control at a certain point in CICS operation. It consists of overlaying a small portion of one of the CICS nucleus programs with instruction to transfer control to the vendor product any time control passes through that point. CICS-WINDOWS uses one “hook,” in the Program Control Program (DFHPCP). This was necessary because IBM has not yet provided a Global User Exit at entry to the PCP.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

HOT KEY
A “hot” key is the term used to indicate a PF or PA key which, when pressed, will activate some CICS-WINDOWS function rather than perform its normal function in CICS. A “hot” key always overrides the application program running in a virtual terminal. In CICS-WINDOWS various keys can be assigned to be “hot” keys. They are: the Toggle Forward key, the Toggle Backward key, direct session keys and the window key. You can use a minimum of one hot key, and more if enough PF/PA keys are available.

PHANTOM SESSION
The Phantom Session is a temporary session for CICS-WINDOWS internal use only. It is used for functions such as displaying the Control Window and delivering messages generated by the Message Broadcasting facility of CICS-WINDOWS. This session is made available when needed by CICS-WINDOWS and disappears once the need for this session is resolved.

POPUP WINDOW
Popup window mode is the method of using windows where there is a DOMINANT window and one or more RECESSIVE windows and a maximum of 9 windows that are free-floating (that is, the window is not tied to a corner of a screen). Only the dominant window is active (the ”A” is provided for you), and only the dominant window is unprotected. That is, data can be entered in the dominant window only.

PROGRAM
This is the user or vendor-written set of instructions to the computer which is invoked in CICS by entering a transaction code.

PSEUDO CONVERSATIONAL
This is a method of programming whereby a program, when needing more input from the operator, save enough information on some disk file in order to find its place again, then releases the CICS resources in use and goes away. This is contrasted to “conversational” programs which do not release their resources across screen displays.

PSEUDO TERMINAL ID
This refers to the feature of CICS-WINDOWS whereby a unique four-character terminal identifier can be assigned to each virtual terminal. Pseudo Terminal IDs are not always in use, depending on your initialization selections, but if they are, virtual terminal number one always uses the real terminal ID, while virtual terminals number 2 through 9 use a different 4-character ID which is unique among all other real and Pseudo Terminal IDs in the system.

PULL-DOWN MENU
A pull-down menu is the menu that drops down from the ACTION BAR in order to invoke a function (by tabbing to the desired option and pressing ENTER.)

REAL TERMINAL
This term refers to virtual terminal number one (see VIRTUAL TERMINAL). It is used in connection with Pseudo Terminal IDs, since virtual terminal number one always contains the real terminal ID.

REAL TERMINAL ID
The original four-character identification of the physical terminal as it is known to CICS.

RECESSIVE WINDOW
A recessive window is one which is inactive, is protected so that no data may be entered into it, and may be overlapped by the dominant window. When using “variable” or “popup” window mode there will always be at least one recessive window.

SCROLLING
The technique of “shifting” data up, down, right or left when it is displayed in a window, in order to see all of it.

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

SESSION This term is used interchangeably with VIRTUAL TERMINAL. In the literal sense, it refers to a virtual terminal when an application program has been started in it, as opposed to one where no transaction is active.

SPLIT SCREEN This term is used interchangeably with WINDOWS. It refers to the ability of the CICS-WINDOWS user to divide the screen into two or more parts, each part (window) displaying data from one SESSION.

TASK The term TASK can usually be used interchangeably with TRANSACTION. Strictly speaking, a task consists of everything that happens in the computer from the time that a transaction code is entered until a display is received on the screen. There could be several programs involved in a single task.

TERMINAL See PHYSICAL TERMINAL

TOGGLING The process of moving from one virtual terminal to the next by pressing the designated function key.

TRANSACTION

A transaction is a function performed on a terminal which consists of entering some data, pressing the ENTER key or some program function key and receiving a display on the screen. A transaction is initiated by entering a 1 to 4 character code, known as the transaction code, or sometimes by simply pressing one of the PF or PA keys. Generally speaking, there is a transaction code associated with each program.

VIRTUAL TERMINAL

A virtual terminal is a logical extension of a physical terminal. The term refers to the ability of CICS-WINDOWS to redefine a physical terminal as if it were two or more (up to nine) physical terminals. Each virtual terminal can do anything that the physical terminal can do. You move from virtual terminal to virtual terminal by pressing a “toggle” key. Thus, a physical terminal defined by CICS-WINDOWS as, let’s say, four virtual terminals would be like having four physical terminals setting on your desk.

WINDOW A window is a portion of the physical screen display. There can be from 2 to 9 windows simultaneously displayed on the screen. Windows may vary in size and shape. The are bordered with a row of under-scores at the top and bottom and a column of vertical bars on the left or right side. A window can be empty (clear) or it can contain (display) data from the SESSION which is active in a VIRTUAL TERMINAL.

WINDOW COMMAND LINE

This is the top window bar of a window. It contains the window command area, a 4-character space in the command line where a window command can be entered.

WINDOW BAR The horizontal or vertical border of a window.

WINDOW MODE

The method of operation where the screen is divided into multiple parts (windows). A transaction is initiated in a window by designating an ACTIVE window and entering the transaction code as if it were a full screen.

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

INDEX

A

abend

AKCS, 203, 208, 266, 282
APCT, 257
ATNI, 184, 379
ATZW, 269
definition of, 404
VOLLIE program, 247

ACF2

considerations for, 258
sample exit for, 323
terminal identification, 318

action

functions of
POPUP window mode, 40

action bar

functions of
Auxiliary Functions transaction menu, 200
general operation of, 67

activation

automatic start, 302
CICS-WINDOWS in PLT, 257
from user program, 302
of CICS-WINDOWS 3.1, 51
of the Demonstration Program, 18
with interactive interface, 274

activation prompts

number of terminals, 280
Pseudo Terminal ID's, 281
TOGGLE key, 280
window key, 280
windowing support, 281

AKCS abend

purge at signon/signoff, 203
timeout purge, 208
WNDO,PURGE, 282

APCT abend

disabled program, 257

application interface

commands used in, 296
description of, 294
examples of, 298
functions of, 294
invocation of, 296
program invocation, 300
return codes of, 302
streaming of commands, 294

Application Startup table

commands of, 233
description of, 231
example of, 232
fields of, 231

ATNI abend

READ BUFFER, 205
TIOAL, 184, 379

attribute code

in help menus, 135

ATZW abend

not using pseudo ID's in MRO, 269

AUTODEF window, 99

auto-define

changing the window, 93
field window, 92

Auto-Init table

commands of, 226
description of, 221
fields of, 221

auto-initiated transactions

use of with toggling, 259

Automatic transaction starting. *See* Application Startup table

Autostart table

reference, 216

Auxiliary Functions transaction

Action Bar functions, 200
commands of, 199
description of, 199
invocation of, 199

B

BACKOUT command

use of, 257

backward switch key

changing, 55, 217

backward toggle key

changing, 55
definition of, 55, 217

built-in functions

entry points, 308
linkage conventions, 310
types of, 308

C

CEMT

new copy command, 257

chaining

of help displays, 142

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

CICS

requirements of, 184, 379
command codes

AUTODEF, 93
BACKOUT, 257
COPY, 70
CPROFF, 279
CPRON, 279
CUT, 140, 146
DEBUG, 279
DEMO, 18
DOWN, 24
INIT, 279
INQ, 51
LEFT, 24
OFF, 60
ON, 51, 302
PASTE, 70, 140, 146
PURGE, 282
RENUM, 129
RIGHT, 24
STACK, 70
START, 282
STOP, 282
SWITCH, 29
SYS, 61
TEMP, 283
UNP, 71
UP, 24
WD, 28
WL, 28
WR, 28
WS, 29
WU, 28
X, 26
commands

activate compression, 279
activating CICS-WINDOWS, 51
application interface, 296
changing window size, 28
cut and paste, 69
display acctive users, 61
entering window mode, 23
expanding a window, 26
miscellaneous commands of WHLP transaction, 125
restart CICS-WINDOWS, 282
running the demonstration, 18
scrolling, 24
special-purpose, 278
summary of, 373
switch to Temporary Storage, 283
switching windows, 28
terminate a user, 282
terminate CICS-WINDOWS, 60
terminate compression, 279
compression

activating, 279
deactivating, 279
display of statistics, 62
exclusion
of terminals, 197
of transactions, 197
control character

definition of, 202
description, 64
use of, 66
control key

changing, 52
definition of, 213
use of, 66
Control Menu, 66

conversational tasks

display of, 61
converting the WNDOTBL macro, 194

copy command

description of, 70
CPROFF command

description of, 279
CPRON command

description of, 279
cursor

in help menus, 135
with multiple windows, 144
cursor-select key

use of, 261
customization

dynamic tables of
Application Startup, 231
Auto-Init, 221
File, 210
Signon/Signoff Transactions, 239, 241
single occuring transactions, 247
single occurring programs, 249
stopped transactions, 251
Terminal Compression, 243
terminal exclusion, 234
transaction compression, 245
Transaction Exclusion, 237
User Options, 201
user profiles, 212
window mode stopped transactions, 253
relationships of dynamic tables, 198
WNDOTBL
installation of, 196
WNDOTBL macro
description of, 195
operands of, 195
cut and paste

commands of, 69
description of, 146
examples of, 76
field propagation, 73

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

cut and paste feature, 69

D

data compression. *See* compression

DEBUG command

description of, 279
delp definition directory, 123

Demo Command

description of, 18
demonstration

description of, 18
DFHZNEP

deactivating CICS-WINDOWS, 315
direct session keys

changing, 54
definition of, 216
description of, 22
directional options

in help mode, 101, 104, 108, 110, 140
directory for help definitions, 123

dominant window

activating, 29
color, 52, 56, 213, 219
highlighting, 52, 56, 213, 219
DOWN command

description of, 24
dummy transactions

description of, 135
illustration of, 135
dynamic customization. *See* customization

E

ED command

description of, 28
editor. *See* text editor

end user

use of help-WINDOWS, 137
error conditions

deactivating CICS-WINDOWS, 315
error messages. *See* messages

exclusion

of terminals from CICS-WINDOWS, 197, 234
of terminals from compression, 243
of transactions from CICS-WINDOWS, 197, 237
of transactions from compression, 245
of transactions from toggling, 237
exits. *See* user exits

extended attribute support

changing, 52
defining, 213

F

field directory, 124

File table

description of, 210
fields of, 210
forward switch key

changing, 55, 217
forward toggle key

changing, 55
definition of, 217
description of, 20
FULL attribute

description of, 41
example of, 48

G

generic entry

compression exclusion, 244, 246
Sign-off control, 240
single-occurring tables, 248, 250
Terminal Compression table, 244
Terminal Exclusion table, 234
Transaction Compression table, 246
Transaction Exclusion table, 238
Transaction Stop table, 252
Window Mode Stopped Transactions table, 254
graphic escape

changing, 53
defining, 214

H

Help (on-line), 65

help access key

changing, 53
defining, 89
definition of, 214

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

Help command, 65

help definition key

defining, 89
help displays

chaining, 142
creation of, 88
exiting from, 145
help menus, 135
retrieval of
screen level, 141
transaction level, 138
help menu

creation of, 135
defining, 135
help mode

description of, 137
HIDE attribute

description of, 41
example of, 47
horizontal windows

specification of, 281
specifications of, 218

I

ICCF

considerations for, 259
considerations in VSE, 276
use of PURGE with, 282
ICON attribute

description of, 40
example of, 47
ICS

use of CICS-WINDOWS with, 50
importing help text, 133

inactivity lock. *See* terminal lock

INIT command

description of, 279
initiation of CICS-WINDOWS. *See* activation of

INQ command

description of, 51
installation

DOS/VSE, 380
requirements for, 184, 379
special notes, 387
summary, 387
tape contents, 185, 380
interactive interface

activating CICS-WINDOWS, 274
definitions for, 217
installation for, 274
methods of operation with, 273
overhead considerations, 276
performance condiserations for, 256
use of, 273
interface routines

deactivation at auto logoff, 316
distributed with product, 323
node error deactivation, 315
obtaining the real terminal ID, 318
performing commands, 294
reasons for, 293
types of, 293
using the Active User table entry, 318
ISC

considerations for, 269
dummy TCT entries, 270

J

JCL

load install program DOS/VSE, 380
load source members, 185
product password install, 189
running the install program DOS/VSE, 380
job control

product password install, 189

K

keys. *See* PF keys

keywords

CICS, 185
CSDLIB, 186
DIRECTORY, 185
FSRVFILE, 186
HELPSCRN, 186
HELPTEXT, 187
JUGLFILE, 187
LINES, 187
MODE, 187
of STSINST program, 185
OPSYS, 188
PRODUCT, 188
VTWOFILE, 188
WNDOFILE, 188

L

LEFT command

description of, 24

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

LOCATION window, 106

lock feature. *See* terminal lock

M

menu editor. *See* text editor

menu generation

editor of, 153, 173
menu definition, 159
commands of, 163
display of, 159
fields of, 159
menu directory, 150
commands of, 151
display of, 150
fields of, 150
user exit, 290
installation of, 290
MENUEXIT

sample program, 323
message broadcasting

message definition
commands of, 171
display of, 169
fields of, 169
message directory, 166
commands of, 167
display of, 166
fields of, 167
message editor. *See* text editor

messages

WA prefix, 326
WB prefix, 332
WD prefix, 336
WH prefix, 340
WI prefix, 354
WM prefix, 355
WMR prefix, 357
WN prefix, 358
WNDO prefix, 359

migration

of the WNDOTBL, 194
MRO

considerations for, 269
dummy TCT entries, 270
use of CICS-WINDOWS with, 50
multiple windows

description of, 142
example of, 142
exiting, 145

N

nested help

description of, 144
example of, 144
exiting, 145
retrieval of, 144
new copy command

for windows programs, 257

O

OFF command

description of, 60
ON command

description of, 51
imbedded in sign on, 302
use of, 19
optimization. *See* compression

P

panning. *See* scrolling

elimination of, 140
in help mode, 140
left, 140
right, 140

password

for product
description of, 189
fields of, 191
installation of
DOS/VSE, 189
MVS, 190
PASTE command

description of, 70, 146
use of, 140
performance

considerations for
conversational tasks, 255
disabling read buffer, 205, 215
general, 255
high activity, 255
READ BUFFER, 255
PF keys

definition
Function Key Assignment Display, 58
definition of
backward toggle, 55
backwards toggle, 217
control, 52, 213
forward toggle, 217
help access, 53
helpaccess, 214
scrolling keys, 54, 215, 216
sessions direct, 54, 216
window key, 56
window switch, 57, 220
description of
backward toggle, 22
direct session, 22
forward toggle, 20
scrolling, 25
switch keys, 31
description of window key, 23
selection of
toggle forward, 280
window key, 280

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

PLT

initialization of CICS-WINDOWS, 257
POPUP windows

action bar functions, 40
examples of, 48
programmer interface

command level example, 306
description of, 302
macro level example, 305
programs

calling CICS-WINDOWS from, 302
WNDODEMO, 18
prompts. *See* activation prompts

pseudo terminal IDs

considerations for, 258
controlling generation of, 287
definition of, 221
disabling, 201
display of, 62
reasons for use, 258, 281
selection of, 280
PURGE command

description of, 282

R

read buffer

disabling, 205, 215
impact on view function, 63
performance considerations, 255
reformat program (for help text), 133

renumbering text records, 129

RIGHT command

description of, 24

S

sample programs

MENUEXIT, 323
on installation tape, 323
WNDOACF2, 323
WNDOCSSF, 323
WNDOCSSN, 323
WNDOINT3, 324
WNDOINT5, 324
WNDONEPC, 324
WNDONEPM, 324
WNDOPURG, 324
WNDORSDD, 324
WNDORSDM, 324
screen displays

activation prompts, 280
Active Message Status, 178
Application Startup table, 231
Auto-Init table, 221
cut and paste examples, 76, 79, 81
File table, 210
help-window, 141
menu definition, 159
menu directory, 150
message definition, 169
message directory, 166
message reception, 182
multiple windows, 142
nested help, 144
operator sign-on, 19
Signoff Transactions table, 239
Signon Transaction table, 241
Single Occurring Programs table, 249
Single Occurring Transactions table, 247
Stopped Transactions table, 251
system display, 61
Terminal Compression table, 243
Terminal Exclusion table, 234
Terminal Status, 180
Transaction Compression table, 245
Transaction Exclusion table, 237
user configuration, 19, 51
User Function Assignment, 58
User Options table, 201
User Profile table, 212
Window Mode Stopped Transactions table, 253
windowing examples, 41
screen identification window, 109

scrolling

commands, 24
description of, 24
illustration of, 25
operation with keys, 25
scrolling columns

defining, 216
scrolling keys

changing, 54, 215
description of, 25
scrolling rows

defining, 216
selection at initiation. *See* activation prompts

session termination

options at, 206, 263
sessions

defining number of, 214

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

displaying in windows, 23
integration of, 232
sign off/sign on

at terminal lock, 267
calling CICS-WINDOWS at, 302
deactivating CICS-WINDOWS, 316
options at, 263
sign-off

while CICS-WINDOWS is active, 60
Signon/Signoff Transactions table

commands of, 240
description of, 239, 241
fields of, 240, 242
Single Occurring Programs table

commands of, 250
description of, 249
fields of, 250
Single Occurring Transactions table

commands of, 248
descritpion of, 247
fields of, 248
split (text record), 114

STACK command

descriptio of, 70
START command

description of, 282
statistics (compression)

display of, 62
STOP command

description of, 282
Stopped Transactions table

commands of, 252
description of, 251
fields of, 252
window mode
description of, 253
storage requirements

of CICS-WINDOWS, 377
STS0100

link-edit of DOS/VSE, 380
STSINST

keywords of, 185
STSPASS

assembly of, 189
description of, 189
fields of, 191
SWITCH command

description of, 29
switch keys

changing, 57
backward, 55, 217
forward, 55, 217
definition of, 220
description of, 31
SYS command

description of, 61
system directory, 124

System Display

example of, 61
fields of, 61

T

table of contents

creating help for, 131
TEMP command

description of, 283
terminal

virtual
selection of ID, 280
selection of number, 280
Terminal Compression table

commands of, 244
description of, 243
fields of, 244
terminal errors

deactivating CICS-WINDOWS, 315
Terminal Exclusion table

commands of, 236
description of, 234
fields of, 234
terminal lock

use of, 267
terminals

exclusion from CICS-WINDOWS, 197, 234
exclusion from compression, 197, 243
text editor

ATTRIBUTES function, 117, 155, 176
color, 117, 155, 176
command field
block commands
copy, 113, 175
deletion, 113, 154, 175
move, 113, 175
copy, 154
line copy, 113, 154, 175
line deletion, 113, 154, 173
line insertion, 113, 154, 175
line move, 113, 154, 175
move, 154
commands of, 120, 158, 177
EJECT command, 114
extended attributes, 117, 155, 176
for help transaction, 111
for menu transaction, 153

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

for message transaction, 173
highlighting, 117, 155, 176
INCLUDE command, 115
insert
STRTFLD/STOPFLD, 114
line editing, 111, 153, 173
SPACE command, 114
STOPFLD command, 115
STRTFLD command, 115
text record split, 114
use of, 153, 173
time-out. *See* terminal lock

toggle keys

changing
backward, 55
direct, 55
forward, 55
definition of
backward, 55, 217
direct, 216
forward, 217
description of
backward, 22
direct, 22
forward, 20
selection of
backward, 55
direct, 55
use of
forward, 20
using cursor-select, 261
toggling

example of, 20
examples of use, 49
exclusion of transactions, 237
in window mode, 26
operation with, 19
transaction codes

dummy, 135
WAUX, 199
WDMO, 18
WHLP, 88
WNDO, 51

Transaction Compression table

commands of, 246
description of, 245
fields of, 245
transaction directory, 123

Transaction Exclusion table

commands of, 238
description of, 237
fields of, 238
transaction key record, 125

chaining, 142
transactions

automatic start of, 160, 231, 232
defining single occurance of, 248
exclusion from CICS-WINDOWS, 197
exclusion from compression, 245
exclusion from toggling, 237
use of auto-initiated, 259

U

UNP command

description of, 71
UP command

description of, 24
upper case translation

resetting, 130
user area

of TCTTE, 203
User Configuration Display

example of, 19, 51
fields of, 52
suppression of, 307

user exits

attach exit
description of, 285
menu generation feature, 290
installation of, 293
pseudo ID generation
description of, 287
pseudo ID generation exit
specification of, 205
types of, 284
User Function Key Assignment Display

description of, 58
User Options table

commands of, 209
description of, 201
fields of, 201
User Profile table

commands of, 220
description of, 212
fields of, 212
protecting, 215

V

vertical windows

specifications of, 218
view function

description of, 63
requirement for, 205, 215
use of, 63
viewing

another terminal screen, 63
virtual terminal

selection of ID, 280
selection of number, 280
VOLLIE program abend

running in multiple sessions, 247

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

VSAM file

defining multiple files, 210

W

WAUX transaction, 199

WHLP directory, 123

WHLP transaction code, 88

wild-card characters

auto-start, 223, 225, 235
compression exclusion, 244, 246
sign-off control, 240, 242
single-occurring tables, 248
terminal, 169, 170
Transaction Exclusion table, 238
Transaction Stop table, 252
Window Mode Stopped Transactions table, 254

WIN command

use of, 23

window action bar, 40

window adjustment commands

description of, 28
limitations of, 28

window command area

description of, 24
use of, 24

window disposition

defining, 219

window key

changing, 56
changing of, 56
description of, 23
selection of, 280

window mode

default mode, 24
entering, 23
toggling while in, 26

Window Mode Stopped Transactions table, 253

commands of, 254
fields of, 254

window size and position

changing of, 28
defining, 219

window switch

keys, 220
changing, 57

window SWITCH

command, 29

windowing

definition of, 56
examples of use, 49
selection of, 280

windows

changing configuration of, 57
configurations of, 23
designating dominant, 29
expanding to full screen, 26
operation with, 23
recessive attribute examples, 48
recessive attributes, 40
selecting configuration, 280
selecting number of, 280

WL command

description of, 28

WNDOACF2

description of, 323

WNDOCSSF

description of, 323

WNDOCSSN

description of, 323

WNDODEMO

description of, 18

WNDOINT3

description of, 324

WNDOINT5

description of, 324

WNDONEPC

description of, 324

WNDONEPM

description of, 324

WNDOPURG

description of, 324

WNDOREFM program, 133

WNDORSDD

descritpion of, 324

WNDOVSP

description of, 274

word-wrap

controlling, 129
description of, 140

WR command

description of, 28

WS command

description of, 29

WU command

description of, 28

X

X command

description of, 26

X-doc record, 127

© Copyright 1995-2004 UNICOM Systems, Inc. All Rights Reserved. GP50-0420-2 CICS-WINDOWS V4.2.0

© Copyright 1995-2004 UNICOM Systems, Inc. All rights reserved. GP50-0420-2 CICS-WINDOWS V4.2.0